

Numerical Analysis Exam with Solutions

Richard T. Bumby

Fall 2000

June 13, 2001

You are expected to have books, notes and calculators available, but computers or telephones are not to be used during the exam. You should check that you have a complete exam. There are 6 problems on 3 pages (printed single sided). All work for this exam is to be done in the blue books provided. You may keep the question paper.

1. (35 pts.) Let

$$g(x) = 0.1 + 0.6 * \cos(2x).$$

Investigate the possibility of solving $x = g(x)$ by iterating g (as in Algorithm 2.2). If you find an interval for which the hypotheses of theorems 2.2 and 2.3 are satisfied, use it to estimate the number of iterations needed to get 10 decimal place accuracy for the fixed point. In any case, show that there is only one solution of $x = g(x)$. If convergence is uncertain (or slow), try a different rootfinding procedure. Describe your chosen procedure and apply it to give 10 decimal place accuracy. Can you now find an interval satisfying the hypotheses of theorems 2.2 and 2.3?

Solution

2. (35 pts.) Let $y \geq 1$ be given as a function of $x \geq 0$ by

$$y^3 - y = x.$$

(The restriction on x and y guarantees that a unique root of the cubic is selected for all admissible values of x). The inverse function theorem allows us to tabulate some values of this function and its derivative at irregularly spaced values of x (chosen to give nice values of y). Here is such a table.

x	y	dy/dx
0.000	1.0	1/2.00
0.231	1.1	1/2.63
0.528	1.2	1/3.32
0.897	1.3	1/4.07
1.344	1.4	1/4.88
1.875	1.5	1/5.75
2.496	1.6	1/6.68
3.213	1.7	1/7.67

Use this to construct a divided difference table to give an appropriate interpolating polynomial of degree 2 for approximating the value of y when $x = 1$. What value do you get for y ? To test the accuracy, compute $y^3 - y$. What does this say about the accuracy of the value of y ? Now, use values of y and dy/dx at appropriate values of x to construct a divided difference table to give a Hermite cubic to approximate the value of y when $x = 2$. Test the accuracy of this as before.

Solution

3. (30 pts.) By using the Taylor series for $\cos(x)$, one can prove that

$$\frac{1 - \cos x}{x^2} = \frac{1}{2} - \frac{x^2}{24} + \frac{x^4}{720} - \frac{x^6}{40320} \cos \xi$$

for some ξ between 0 and x . For which x will this formula compute the function accurately to within 10^{-10} ? Apply the formula to evaluate the function at $x = .1$, $x = .05$, $x = .01$, $x = .005$ and $x = .001$ to the accuracy of your calculator. Compare the answers to the value obtained by subtracting the value given by the \cos key on your calculator from 1 and dividing by x^2 . They should be (slightly) different. Explain the difference.

Solution

4. (30 pts.) Here is a portion of a table of a function.

x	y
0.3	1.531475
0.4	1.402252
0.5	1.284457
0.6	1.177036
0.7	1.079035

Approximate the derivative of this function at each of the given values of x using the best *three-point* numerical differentiation rule for each point. **Show the formula being used as well as the answer.** Also use the *five-point* rule of equation (4.6) of the textbook to obtain what should be a much better estimate of the derivative at 0.5. Assuming that this is a much better estimate, how small would you expect the **truncation** error to be if you used the three point formula based on values of the function with x increments of 0.01 instead of 0.1? If the function was still tabulated to 6 decimal places, what would the **round-off** error be?

Solution

5. (35 pts.) Devise a plan to find

$$\int_{1.5}^2 \frac{e^x}{x} dx$$

to 8 decimal places. Begin with the simplest form of the trapezoidal rule and repeatedly half the step size in the **composite** trapezoidal rule. Use this as a basis of a **Romberg method** to develop higher order rules. Even if the Romberg method does not yield a quick answer, it should allow you to estimate the number of times you must halve the step size to get an acceptable answer with the composite trapezoidal rule and with the composite Simpson's rule. Give the step sizes needed for these rules. If excessive computation would be required to give a suitable value of the integral, describe how the problem would be done on a computer. (If your calculator has a built-in Simpson's rule that you have used and you have determined that it will give a suitable answer, use it and describe how you entered the data for this method and the answer obtained. Otherwise, describe your choice of the best method for obtaining an answer and any programming considerations.)

Solution

6. (35 pts.) Consider the differential equation

$$\frac{dy}{dt} = \frac{y}{t} - \left(\frac{y}{t}\right)^2$$

with $y(1) = 1$. Select a **second-order method** (either a Taylor method or one of the three second-order Runge-Kutta methods described on pages 279-280 of the textbook) and compute **two steps** with $h = 0.1$. Then perform a single step of the fourth-order Runge-Kutta method with a step size of $h = 0.2$. (Since the solution of the equation is known, we have that the value of y when $t = 1.2$ should be 1.01495231404. How accurate are your results?) Outline a program for solving this equation for $1 \leq t \leq 2$ to six decimal place accuracy. Include a test that will check the accuracy of the answer that does not depend on having an exact solution to the equation.

Solution

Solution of Problem 1

For iteration of

$$g(x) = 0.1 + 0.6 * \cos(2x)$$

to converge, we need

- (a) An interval I that is mapped into itself.
- (b) A constant $k < 1$ such that $|f'(x)| \leq k$ on I .

These conditions are easy to satisfy separately. For example, since

$$-1 \leq \cos(2x) \leq 1$$

for all x , we have

$$-0.5 \leq g(x) \leq 0.7,$$

and since $g(x)$ has a local maximum at $x = 0$,

$$0.2 < g(0.7) \leq g(g(x)) \leq g(0) \leq 0.7.$$

Also, $g'(x) = -1.2 \sin(2x)$, so (b) will be satisfied if

$$|x| \leq 0.5 \arcsin(1/1.2) \approx 0.49255539.$$

However, none of the intervals mentioned so far satisfy *both* conditions. We now attempt to merge the conditions to find a suitable interval.

There is nothing to be gained by considering an interval that is not contained in $[0.2, 0.7]$, since all values of $g(g(x))$ are in this interval. We first note that $g(x)$ is decreasing on this interval (by considering the derivative). If the endpoints of such an interval I are taken into I by g , (a) will be satisfied, but (b) requires that the right endpoint not be larger than $0.5 \arcsin(1/1.2)$, so the left endpoint cannot be smaller than the preimage of this value under g , which is approximately 0.4288. If this were not smaller than our chosen right endpoint, we would have shown that no such interval I could exist. As it is, any choice of the interval I must be contained in the interval that we have just determined. We need a little more since we need to find a $k < 1$ that bounds $|g'(x)|$. If we arbitrarily take $x_0 = 0.45$, we find that $g(x_0) < 0.473 = x_1$

and $g(x_1) > x_0$, so we can take $I = [x_0, x_1]$. Finally, $|g'(x)|$ is increasing on this interval, so we can take $k = |g'(x_1)|$, which is slightly less than 0.9733, and we find that $(0.9733)^n < 10^{-1}$ for $n \geq 85$, so that 85 iterations will give one new decimal place of accuracy.

With patience, this will find the fixed point, but it is too slow. Only 30 bisection steps would be needed to find where $f(x) = x - g(x)$ is zero, starting from the interval I just constructed (the construction shows that f changes sign of I). Since the derivative of f is easily determined, we could use Newton's method to find where $f(x) = 0$. Alternatively, the iteration of g could be accelerated by Steffensen's method. Both of these are quadratically convergent, but we can get off to a good start by starting the iteration at $x_0 = 0.45$. The $p_0^{(i)}$ for Steffensen's method are

0.45
0.461736559735
0.461785307032
0.461785307873

and for Newton's method, the values are

0.45
0.461838181555
0.461785308906
0.461785307873

and the last value listed is seen to be a fixed point of g , and hence a zero of f to the full calculator accuracy of 12 decimal places. When we know the fixed point x_∞ , the k for iteration of g on a small interval around x_∞ will only be slightly larger than $|g'(x_\infty)|$, so only about 53 iterations are needed to get one new decimal place of accuracy.

Note the convention of using f to denote a function whose zero is to be found, and g a function whose fixed point is to be found. The two problems are equivalent, but one must modify the function when going from one formulation to the other. Also, a calculator must be set in radian mode to calculate the mathematically recognized trigonometric functions.

A small change in the definition of the function g can give a function for which iteration does not converge. For example, for

$$g_2(x) = 0.2 + 0.6 * \cos(2x)$$

iteration will lead to a cycle that alternates between 0.35176928145 and .65753469269. Other methods will locate the unique fixed point at 0.511986175381.

Solution of Problem 2

Divided differences give the easiest determination of interpolating polynomials. One records the given data at the left side of a table and computes new columns as differences of consecutive elements in the previous column divided by an appropriate difference in the argument. For example, $f[a, b, c] = (f[b, c] - f[a, b]) / (c - a)$. To evaluate the interpolating polynomial at a new argument, one uses *nested multiplication* (a more general form of Horner's method that appears in Theorem 2.18). An easy way to do the calculation is to write the new argument x on a new line of the divided difference table, which leads to a new diagonal

of divided differences. To find these, copy the rightmost element and fill in the remaining values from right to left. When you are done, you will have the value of the interpolating polynomial at x . A typical step is given by rewriting the divided difference equation as, for example, $f[c, x] = f[b, c] + (x - b)f[b, c, x]$. Although it is common to arrange the arguments in increasing order when working with divided differences, this is not necessary. Indeed, the value of a divided difference is independent of the order of its arguments. However, when there are repeated arguments, the equal arguments should be kept together to allow the derivatives of the function to be inserted near the beginning (i.e, the left side) of the table. Otherwise, it would be necessary to identify the derivative of a divided difference whenever the ends of a span of argument belong to equal arguments.

Although a large table was given, only a small number of points near the desired value of x are used in the computation. Three points are used to get a quadratic polynomial. For example, to approximate $f(1)$, we can use 0.528, 0.897, and 1.344. Here is the table with both the divided differences and the nested multiplications.

x	y	yy	yyy
0.528	1.2		
0.897	1.3	1/3.69	
1.344	1.4	1/4.47	-0.0579422836943
1	1.32509587091	0.217744561311	-0.0579422836943

In this table, the x and y values in the upper part of the table are known exactly, and exact expressions used in the yy column of first divided differences. For this table, I used a form of the expression that used these exact expressions; when I allowed the calculator to use its (rounded) version of the first divided differences before subtracting, the value calculated as the second divided difference differed in the last decimal place shown, but this had no effect on the later steps of the calculation.

To test this, we compute $y^3 - y - 1 \approx 0.0016$, which measures how close we came to solving the equation. To turn this into an estimate on the error in calculating y , we note that $dx/dy = 3y^2 - 1$, which is approximately 4.27 when $y \approx 1.325$. Dividing the error in x by dx/dy gives an estimate on the error in y of less than 4×10^{-4} .

In principle, one could use the standard error term in Theorem 3.3 to bound the error. However, the function giving y in terms of x is defined as an inverse function. First derivatives of inverse functions have nice expressions, but the higher derivatives become cumbersome. In this example, we have

$$\begin{aligned}
 \frac{dy}{dx} &= (3y^2 - 1)^{-1} \\
 \frac{d^2y}{dx^2} &= -(3y^2 - 1)^{-2} \cdot \frac{d}{dx}(3y^2 - 1) \\
 &= -(3y^2 - 1)^{-2}(6y) \frac{dy}{dx} \\
 &= -(3y^2 - 1)^{-2}(6y)(3y^2 - 1)^{-1} \\
 &= -(3y^2 - 1)^{-3}(6y)
 \end{aligned}$$

which must be differentiated in a similar way to obtain the third derivative. Since an accurate estimate of the resulting expression may not be worth the effort, this approach may fail to recognize the accuracy that has been demonstrated by other means.

One thing that we do get from the error term in Theorem 3.3 is that the error is essentially the product of a fixed quantity with the cube of the step size used in forming the table. (This takes a little work, since it is differences in values of x that appears in the theorem, and we want to measure things in terms of differences in values of y . However, the ratio of these is approximated by the derivative near the point where the function is evaluated, and this is a quantity that is essentially fixed.) Repeating this with a step size of .01 in y could be expect to reduce the error by a factor of 1000 to 4×10^{-7} .

If the points were not chosen close to intended value the factor

$$(x - x_0)(x - x_1)(x - x_2)$$

in the error term will no longer be a product of three small numbers, which is the feature of the error term that shows that interpolation can provide a good approximation, and that it gets significantly better with a smaller step size.

The second part of the question calls for using a Hermite cubic interpolation in a different part of the table. The analysis is the same, so we only present the divided difference table and computation of the function. Hermite interpolation allows us to use only the values that immediately surround the desired value, with each used twice.

x	y	yy	yyy	$yyyy$
1.875	1.5			
1.875	1.5	1/5.75	-0.0207446822304	
2.496	1.6	1/6.21	-0.0182447616921	0.00402563693768
2.496	1.6	1/6.68	-0.0177415570749	0.00402563693768
2	1.52138379609	0.158500411111		

For $y = 1.52138379609$, we find $y^3 - y - 2 = 0.00002430592$, and combining this with a value of dy/dx of approximately 5.95 at this value of y , we find that the error in y is slightly more than 4×10^{-6} . This is a fourth order approximation, so a table based on a step size of 0.01 in y would lead to an error of about 4×10^{-10} .

Solution of Problem 3

In the given expression, the term $x^6 \cos \xi / 40320$ represents the error when the function on the left is approximated by the polynomial consisting of the first three terms on the right. Since $|\cos \xi| \leq 1$, this error term is bounded by $|x^6 / 40320|$ for all x . This is less than 10^{-10} for $|x| < 0.12615953722$. Thus, for all x at which we were asked to consider, the polynomial will be represent the function to 10 decimal places. In addition, there will be no cancellation of the terms of the polynomial in this range, so roundoff is entirely determined by the register size of the calculator, which is more than 10 places on current calculators. Thus, any larger difference from the value given by direct calculator evaluation of the expression on the left can only be due to the calculator being unable to evaluate that expression accurately. Here are the requested

values, and a few more, as obtained on a Hewlett-Packard 48G.

x	polynomial	expression
0.1	0.499583472222	0.4995834722
0.05	0.499895842014	0.499895842
0.01	0.499995833347	0.49999893
0.005	0.499998958334	0.49999896
0.001	0.499999958334	0.5
0.0001	0.49999999584	0.5
0.00001	0.49999999996	0.5
0.000005	0.49999999999	0.48
0.000003	0.5	0.444444444444
0.000002	0.5	0.5
0.000001	0.5	0

Looking first at the polynomial, we see that the value is given to 12 decimal places until x is so small (i.e., 3×10^{-6}) that all but the constant term of the series is negligible. However, the direct evaluation of the expression is shown to diminished precision from the very beginning, and the difference already exceeds 10^{-10} for $x = 0.01$. For x between 10^{-3} and 10^{-5} , the formula gives the limiting value, although we know that the function is noticeably smaller. After a brief interval of erratic values, the formula then gives 0, which is *very* wrong. The explanation lies in the roundoff error in the computation of the expression. The calculator has a fixed register size, and everything computed, including intermediate steps must fit in that register. The expression requires the value of $\cos x$ for small x , which is very close to 1. Since only a dozen decimal places can be stored, the stored value becomes essentially $1 - x^2/2$ when x is smaller than 10^{-3} and indistinguishable from 1 when x is less than 10^{-6} . Moreover, only a fixed number of decimal places will be accurate, so the *floating point* representation of $1 - \cos x$ is padded in the interval where it should differ only slightly from $x^2/2$, leading to erratic behavior. Even for values at the beginning of the table, the padding makes the computed value at these values of x appear to be exactly a number with fewer than 12 decimal places.

When you press the `cos` key on your calculator, what you get is only *approximately* the cosine of the number in the display. The approximation is likely to resemble the Taylor series expansion for small x , but it may be modified by some of the methods discussed in Chapter 8 of the textbook. This is good enough for normal applications, but this exercise was designed to expose the limitations of these approximations.

Solution of Problem 4

The three-point numerical differentiation formulas are:

(A) $f'(x_0) \approx (-3f(x_0) + 4f(x_0 + h) - f(x_0 - 2h)) / (2h)$

(B) $f'(x_0) \approx (f(x_0 + h) - f(x_0 - h)) / (2h)$

These are specializations of equation (4.3) of the textbook, which also includes an error estimate. The error for formula (B) appears to be half that of formula (A), so it is preferred when both formulas applied. Thus, in this exercise, formula (A) is used only for $f'(0.3)$ (with $h = +0.1$) and $f'(0.7)$ (with $h = -0.1$). For the remaining values, formula (B) is used (with $h = +0.1$, although $h = -0.1$ gives exactly the same results). Since we are starting with only six decimal place values, and are dividing by 0.2, the results will have only five decimal places.

Here are the results:

x	$dydx$
0.3	-1.34937
0.4	-1.23509
0.5	-1.12608
0.6	-1.02711
0.7	-0.93291

There is also a five-point formula:

$$(C) \quad f'(x_0) \approx (f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)) / (12h)$$

This applies only to finding $f'(0.5)$, and the result (rounded to five decimal places) is -1.12441 . This has a fourth order error term, while formulas (A) and (B) have second order error terms. The instructions in the exercise say to make the reasonable assumption (since the function seems to be well behaved) that formula (C) gives a much better estimate. That is, that the difference between the two estimates is very close to the error in the three-point formula. This estimates the truncation error in the three-point formula to be about 1.7×10^{-3} when $h = 0.1$. Since this has a second-order error term, dividing the step by 10 should divide the error by 100, giving an estimate of 1.7×10^{-5} in the center of the range.

Roundoff error is a different story. The numerator of the formula is only accurate to 10^{-6} because the tabulated values were only given to this accuracy. Dividing by $2h$ gives an error of 5×10^{-6} when $h = 0.1$, but 5×10^{-5} when $h = 0.01$. In particular, the roundoff error may be larger than the truncation error at this step size.

Another approach to estimating the error is to use the fact that four-point divided differences of a function f also provide an estimate on $f'''(x)/6$ in the interval. From the given table, these divided differences are -0.176 and -0.159 . This is only slightly more cautious than our previous estimate.

Solution of Problem 5

Computing *averages* rather than integrals organizes the computation in a way the guards against blunders. When the average has been found, it is multiplied by the length of the interval to get the integral. That approach will be taken, so the tabulated values will be averages and the integral is found by multiplying by 0.5. The same factor is used to convert any approximate average to the corresponding approximate integral, if desired.

The integrand is seen to fairly *smooth*. That is, repeated differentiation, while possibly symbolically troublesome, only leads to linear combinations of e^x/x^k with small coefficients. This makes the problem an ideal candidate for the Romberg method since the observed differences in the estimates will accurately reflect the theoretical truncation error. Roundoff error is bounded by the number of points in the partition times the roundoff error in a single function evaluation. This function can be evaluated to full calculator accuracy (12 decimal places on my calculator), so several thousand points can be used without affecting the requested 8 place accuracy.

Function evaluations enter the Romberg method only in the trapezoidal approximation on the whole interval and composite midpoint rules for each division into 2^n parts for $n = 0, 1, 2, \dots$. The trapezoidal average for $n = k + 1$ is the average of the midpoint and trapezoidal averages for $n = k$. When two successive trapezoidal average T_k and T_{k+1} have been found, a fourth order estimate at level $k + 1$ is obtained as

$$T_{k+1} + \frac{T_{k+1} - T_k}{3}.$$

Successive fourth order estimates are combined in a similar way, using a denominator of 15 instead of 3, to obtain a sixth order estimate. In general, an estimate of order $2j$ is extrapolated with a denominator of $2^{2j} - 1$ to obtain an estimate of order $2(j + 1)$. We record all computations here, using $f(x)$ to name the integrand e^x/x .

$$\begin{aligned}
 \text{Trap} &= \frac{f(1.5) + f(2)}{2} \\
 &= 3.34116038151 \\
 \text{Midpt} &= f(1.75) \\
 &= 3.28834438629 && \text{level 0} \\
 \\
 \text{Trap} &= \frac{3.34116038151 + 3.28834438629}{2} \\
 &= 3.31475238390 \\
 \text{Ord4} &= 3.31475238390 + \frac{3.31475238390 - 3.34116038151}{3} \\
 &= 3.30594971803 \\
 \text{Midpt} &= \frac{f(1.625) + f(1.875)}{2} \\
 &= 3.30147557174 && \text{level 1} \\
 \text{Trap} &= \frac{3.31475238390 + 3.30147557174}{2} \\
 &= 3.30811397782 \\
 \text{Ord4} &= 3.30811397782 + \frac{3.30811397782 - 3.31475238390}{3} \\
 &= 3.30590117579 \\
 \text{Ord6} &= 3.30590117579 + \frac{3.30590117579 - 3.30594971803}{15} \\
 &= 3.30589793964 \\
 \text{Midpt} &= \frac{f(1.5625) + f(1.6875) + f(1.8125) + f(1.9375)}{4} \\
 &= 3.30479004998 && \text{level 2}
 \end{aligned}$$

$$\text{Trap} = \frac{3.30811397782 + 3.30479004998}{2}$$

$$= 3.30645201390$$

$$\text{Ord4} = 3.30645201390 + \frac{3.30645201390 - 3.30590117579}{3}$$

$$= 3.30589802593$$

$$\text{Ord6} = 3.30589802593 + \frac{3.30589802593 - 3.30589793964}{15}$$

$$= 3.30589781594$$

$$\text{Ord8} = 3.30589781594 + \frac{3.30589781594 - 3.30589793964}{63}$$

$$= 3.30589781398$$

Midpt = average of 8 values

level 3

$$= 3.30562073362$$

level 4

$$\text{Trap} = \frac{3.30645201390 + 3.30562073362}{2}$$

$$= 3.30603637376$$

$$\text{Ord4} = 3.30645201390 + \frac{3.30603637376 - 3.30645201390}{3}$$

$$= 3.30589782705$$

$$\text{Ord6} = 3.30589782705 + \frac{3.30589782705 - 3.30589802593}{15}$$

$$= 3.30589781379$$

$$\text{Ord8} = 3.30589781379 + \frac{3.30589781379 - 3.30589781594}{63}$$

$$= 3.30589781376$$

$$\text{Ord10} = 3.30589781376 + \frac{3.30589781376 - 3.30589781398}{127}$$

$$= 3.30589781376$$

level 4

Note that the value of the function at only 17 points in the interval were needed before extrapolation failed to show any change in the value. If further levels were computed, there would be no change in the methods of order 8 or higher. Lower order methods would continue their march to the limit of 3.30589781376 for the average, which gives an integral of 1.65294890688. Since this function is so gently varying, the fourth-order method (which is equivalent to Simpson's rule) can be seen to reach 8 place accuracy in one or two more halvings of the step size, since the error is divided by 16 each time this is done. This is well within the ability of the built-in Simpson's rule on most calculators. The best trapezoidal estimate obtained in our table only differs from its extrapolation by about 1.4×10^{-4} , so seven more halvings of the step size would be needed to reduce this to 10^{-8} , since this error is only divided by 4 by this action. Over 1000 terms must be found and combined, so the roundoff error could cause as many as three decimal places of the computed accuracy to be lost. A 12 decimal place calculator would not be able to guarantee the required accuracy.

Solution of Problem 6

Second-order methods that were discussed in the course are: (1) the Taylor method; (2) the modified Euler method; (3) the midpoint method. Methods 2 and 3 are Runge-Kutta methods reducing to the trapezoidal and midpoint rules, respectively, for approximating integrals when the right side of the equation depends only on t . A fourth method, Huen's method, was mentioned in the textbook, but not discussed in class, so we will show only the results of the first three methods. Since our goal in this problem is 6 place accuracy, results will be shown only to that precision.

First, the Taylor method. The function on the right side of the equation is

$$\frac{dy}{dt} = f(t, y) = \frac{y}{t} - \left(\frac{y}{t}\right)^2.$$

Thus,

$$\begin{aligned} \frac{d^2y}{dt^2} &= \frac{d}{dt} f(t, y) = \left(1 - 2\frac{y}{t}\right) \cdot \frac{d}{dt} \left(\frac{y}{t}\right) \\ &= \left(1 - 2\frac{y}{t}\right) \cdot \left(\frac{1}{t} \frac{dy}{dt} - \frac{y}{t^2}\right) \\ &= \left(1 - 2\frac{y}{t}\right) \cdot \left(\frac{1}{t} \cdot \left(\frac{y}{t} - \left(\frac{y}{t}\right)^2\right) - \frac{y}{t^2}\right) \\ &= -\frac{y^2}{t^3} \cdot \left(1 - 2\frac{y}{t}\right) \end{aligned}$$

In the Taylor method,

$$\begin{aligned} y(t+h) &= y(t) + y'(t)h + \frac{y''(t)}{2}h^2 \\ &= y(t) + h\left(y'(t) + \frac{y''(t)}{2}h\right) \end{aligned}$$

In this problem $h = 0.1$. A table summarizing the computation of two steps is:

t	y	y/t	dy/dt	d^2/dt^2	$y'(t) + y''(t)h/2$
1.0	1.0	1.0	0.0	1.0	0.05
1.1	1.005	0.913636	0.078905	0.627773	0.110294
1.2	1.016029				

Only two entries are made on the third line since those entries are the t and y coordinates of the goal. Comparing with the known solution of 1.014952 shows that the error at $t = 1.2$ is about 1.1×10^{-3} .

In the midpoint method, the tangent at (t_i, y_i) is followed until $t = t_i + h/2$ and the tangent at that point is used to give the direction of the chord from (t_i, y_i) to (t_{i+1}, y_{i+1}) . We denote the slope of the tangent by m_1 and the slope of the chord by m_2 (the usual notation for Runge-Kutta methods would use $k_j = h \cdot m_j$). Here is a summary of that computation.

t	y	y/t	$m_1 = dy/dt$	$t^{(1)} = t + h/2$	$y^{(1)} = y + hm_1/2$	$y^{(1)}/t^{(1)}$	m_2
1.0	1.0	1.0	0.0	1.05	1.0	0.952381	0.045351
1.1	1.004535	0.913214	0.079254	1.15	1.008498	0.876955	0.107905
1.2	1.015326						

Comparing with the known solution shows that the error at $t = 1.2$ is about 3.7×10^{-4} .

The modified Euler method also has $m_1 = dy/dt$, but it follows the line of this slope to $t = t_i + h$, and defines m_2 to be the slope of the tangent at this point. These are averaged to get the m that is used for the official step. Here is a summary of that computation.

t	y	y/t	$m_1 = dy/dt$	$t^{(1)} = t + h$	$y^{(1)} = y + hm_1$	$y^{(1)}/t^{(1)}$	m_2	m
1.0	1.0	1.0	0.0	1.1	1.0	0.909091	0.082645	0.041322
1.1	1.004132	0.912847	0.079557	1.2	1.012088	0.843407	0.132072	0.105814
1.2	1.014714							

Comparing with the known solution shows that the error at $t = 1.2$ is about 2.4×10^{-4} .

In this case, the modified Euler method is accurate enough that a step size of 0.005 would be small enough to give the required accuracy on the given interval. This would use 40 steps, each with two function evaluations. Without further checking, however, one could not be sure that this level of accuracy would persist as far as $t = 2$.

For the fourth order Runge-Kutta method, there are four slope calculations giving m_1, m_2, m_3 and m_4 which are combined to give the official slope $m = (m_1 + 2m_2 + 2m_3 + m_4)/6$. These steps will be written on separate lines for clarity with the average written at the end. The values of t and y for these lines are based on the values on the first line of the block. Only one step of this method has been requested, so there will be a single block followed by the base point for the next interval. Here is a summary of that computation.

step	t	y	y/t	$m = dy/dt$
BASE	1.0	1.0		
1:	1.0	1.0	1.0	0.0
2:	1.1	1.0	0.909091	0.082645
3:	1.1	1.008264	0.916604	0.076441
4:	1.2	1.015288	0.846074	0.130233
AVG				0.074734
BASE	1.2	1.014947		

Now the error is seen to be only about 5.5×10^{-6} , so that a calculation using this method with a step size of 0.1 would more than meet the requested standard of accuracy on this interval since halving the step size can be expected to reduce the error by a factor of 16 in a fourth order method.

One way to get the required accuracy using only comparison of numerical results is to start with something that can be done fairly quickly, but is likely to be moderately accurate. For example, the fourth order

Runge-Kutta method with a step size of 0.2. Save the values obtained for y at $t = 1.0, 1.2, 1.4, 1.6, 1.8, 2.0$. Then repeat the calculation with a step size of 0.1. It takes ten steps to cover the interval this time, but we get new values of y at a collection of values of t including all those that we have saved from the previous computation. Since this is a fourth order method, the difference between these values at the same choice of t should be about 15 times the error of the computation with the finer mesh. Use the largest of these errors as an estimate of error associated with a step size of 0.1. Multiply this by 10^6 and take the fourth root. The next largest integer is the number of pieces our step should be divided into to get the required accuracy. A cautious approach would use two calculations with different step sizes and compare them. The calculations should be performed to sufficient accuracy that the product of number of steps with the size of the roundoff in each step remains less than 10^{-6} .