

# Programming Assignment 4: Graphing a Function

*CS 2073, Computer Programming with Engineering Applications  
Spring Semester, 1992*

Write a Pascal program which will sketch the graph of a function defined in the program.

INPUT: Your program should read an integer  $n$  from the terminal and draw the graph of the function

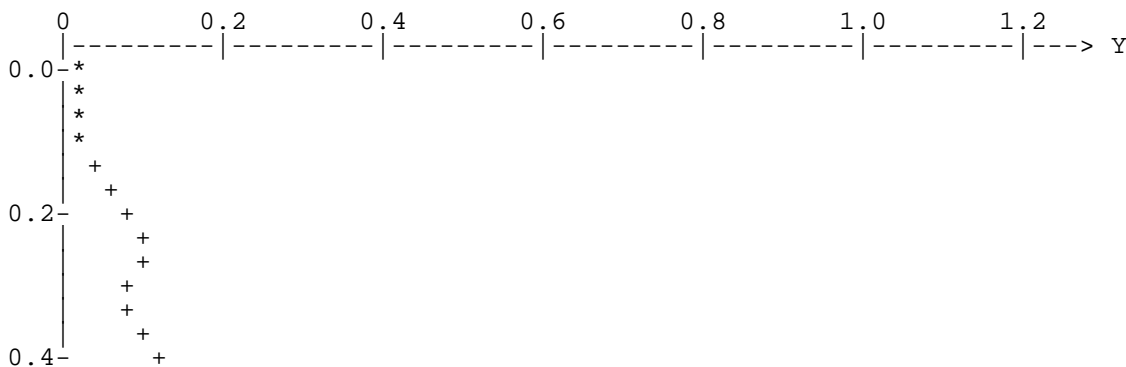
$$f(x) = (2/\pi) [\sin(x) - (1/2)\sin(2x) + (1/3)\sin(3x) + \dots + (-1)^{n+1} (1/n)\sin(nx)], 0 \leq x < \pi.$$

This series converges to the line  $y = (1/\pi)x$ , for  $0 \leq x < \pi$ . For example, for  $n = 3$ , the series takes the form

$$f(x) = (2/\pi) [\sin(x) - (1/2)\sin(2x) + (1/3)\sin(3x)], 0 \leq x < \pi.$$

You should run your program twice, once with  $n = 10$  and once with  $n = 20$ .

OUTPUT: The graph should have the positive  $y$ -axis displayed horizontally and the positive  $x$ -axis vertically (so that one gets the normal configuration by turning the printout 90 degrees counter-clockwise). Your printers use 10 chars/inch horizontally and 6 chars/inch vertically. Suppose we decide to represent 1 unit on the  $y$ -axis by 5 inches or 50 characters, or  $1/50$  units per character. Then 1 unit on the  $x$ -axis must also be 5 inches or  $5 \cdot 6 = 30$  lines, or  $1/30$  units per line. Thus the  $y$ -axis should use 61 characters to display  $y$  values from 0.0 to 1.2, while the  $x$ -axis uses 100 lines to display  $x$  values from 0.0 to 3.333. (The  $x$  values for successive lines are 0,  $1/30$ ,  $2/30$ ,  $3/30$ ,  $\dots$ ,  $100/30$ .) The points of the graph itself should be marked with a "+" as shown below. Both axes should have scaling marks exactly as shown. (The graph shown is only the first portion of what you should produce.)



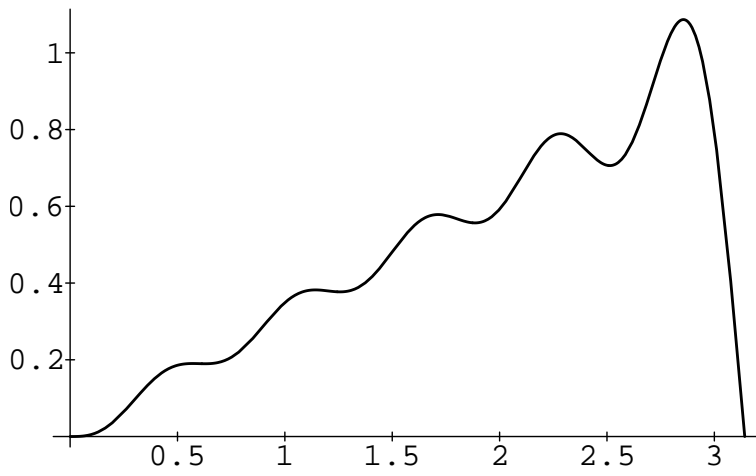
**Notes:**

1. The first  $x$  value is 0, and each subsequent  $x$  value is obtained by adding  $1/30$  to the old value. Each character along the  $y$ -axis represents a distance of  $1/50$ .
2. To decide where to put the "+" character, just multiply the  $y = f(x)$  value by 50.0 and round to the nearest integer. Put "+" that many spaces over horizontally from the  $x$ -axis.
3. The calculation of  $f(x)$  must be done using a Pascal FUNCTION, with a value returned. Pascal PROCEDURES must be used to output the initial  $y$ -axis and to output a line. (The line output procedure must have an integer parameter giving the position for "+".)
4. (Clipping) In case the  $y$  value, when multiplied by 50, gives a value outside the graph, i.e., either  $\leq 0$  or  $\geq 1.2$ , then you should plot a "\*" either in character position 1 (if  $\leq 0$ ) or in position 61 (if  $\geq 1.2$ ). Notice that on the example picture above, the initial numbers were so close to zero that, when truncated, they had to be clipped.

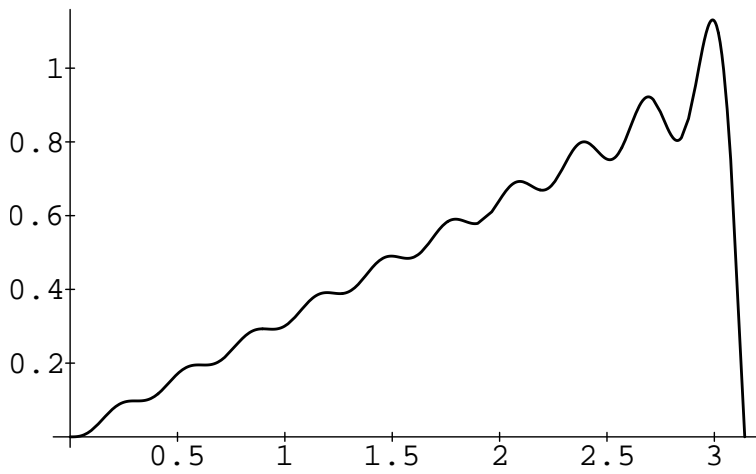
**Extras for experts:**

1. This graph is particularly interesting for values of  $x$  near  $\pi$ . You could try a larger scale so that  $y$  values from 0.8 to 1.2 are plotted, using 60 character positions, or 0.4 inches = 60 chars, or 1 unit = 150 characters, or 1 unit = 15 inches, or  $1/150$  units per character. Along the  $x$ -axis, 1 unit will be 15 inches, or one unit will be 90 lines. Thus the  $x$  values go in steps of  $1/90$ . Plot only  $x$  values near  $\pi$ , say  $x$  from 2.6666 to 3.3333, or  $x$  values from  $240/90$  to  $300/90$  in steps of  $1/90$ . Now to scale the  $y$ -value at each step, you must multiply by 150 and subtract  $0.8*150 = 120$ . Try  $n = 25$ ,  $n = 50$ , and  $n = 100$ .
2. Turbo Pascal has complex features that let you plot points on the screen using the full resolution of the screen, say 640 values horizontally by 480 values vertically on a typical PC screen. This would allow much prettier and clearer graphs to be drawn with your PC. Unfortunately the use of these features involves all sorts of very complex features specific to Turbo Pascal.

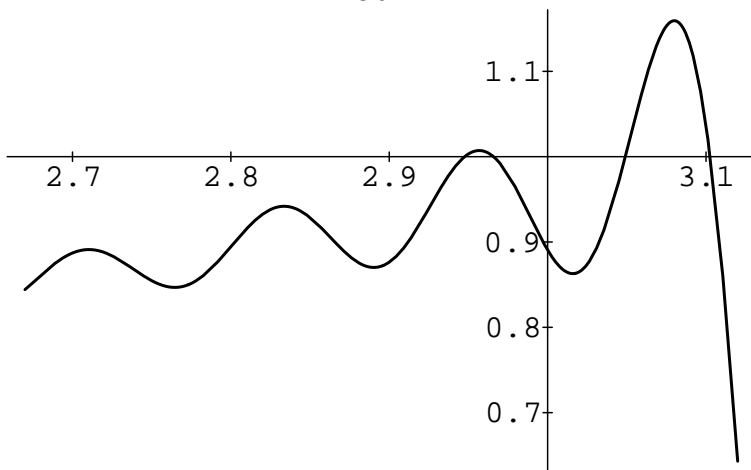
$n = 10$

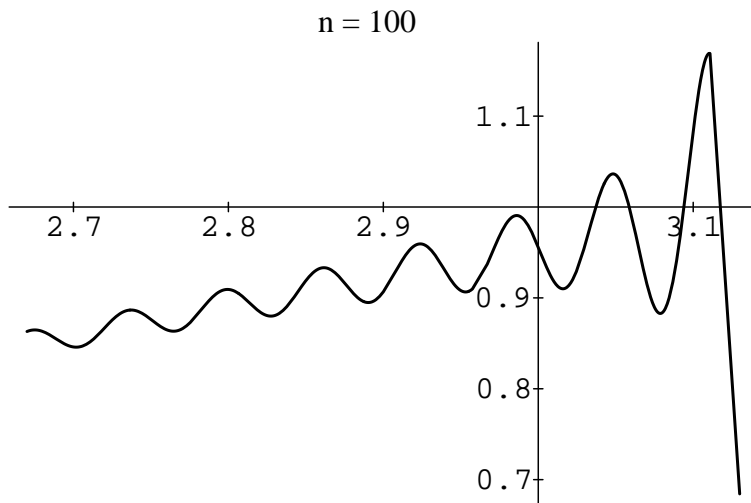


$n = 20$



$n = 50$





```
f[x_,n_] :=  
  Block[{sum = 0.0},  
    For[i = 0, i < n, i++,  
      sum += (-1)^i (1/(i+1)) N[Sin[(i+1)x],10]  
    ];  
    (2/Pi)sum  
  ]
```