Homework 5: Interpolation: Polynomial Interpolation 2 (due on February 28)

---

1. Consider $2n+1$ regularly spaced interpolation nodes $x_{-n}, x_{-n+1}, \ldots, x_{-1}, x_0, x_1, \ldots, x_n$ with $x_k = x_0 + k\,h$, $k = -n, -n+1, \ldots, -1, 0, 1, \ldots, n-1, n$.

   (a) Derive a formula for the interpolation polynomial of order $2n$, using Newton's method and adding points in the order $x_0, x_{-1}, x_1, x_{-2}, x_2, \ldots, x_{-n}, x_n$.

   (b) Derive another formula, adding points in the order $x_0, x_1, x_{-1}, x_2, x_{-2}, \ldots, x_n, x_{-n}$.

   (c) Take the average of the two formulas and show that it is equivalent to Stirling's interpolation formula

$$\begin{aligned} P(x) \quad = \quad & f(x_0) + s\,\frac{\Delta f(x_{-1}) + \Delta f(x_0)}{2} + \frac{s^2}{2}\,\Delta^2 f(x_{-1}) + \\ & \frac{s\,\left(s^2-1^2\right)}{3!}\,\frac{\Delta^3 f(x_{-2}) + \Delta^3 f(x_{-1})}{2} + \frac{s^2\,\left(s^2-1^2\right)}{4!}\,\Delta^4 f(x_{-2}) + \ldots + \\ & \frac{s\,\left(s^2-1\right)\left(s^2-2^2\right)\ldots\left(s^2-(n-1)^2\right)}{(2n-1)!}\,\frac{\Delta^{2n-1} f(x_{-n}) + \Delta^{2n-1} f(x_{-n+1})}{2} + \\ & \frac{s^2\,\left(s^2-1\right)\left(s^2-2^2\right)\ldots\left(s^2-(n-1)^2\right)}{(2n)!}\,\Delta^{2n} f(x_{-n})\;, \end{aligned} \tag{1}$$

   where $s = (x - x_0)/h$, and $\Delta f(x) = f(x+h) - f(x)$.

   (d) Check each of the three formulas using $n = 2$, $h = 1$, and $f(x) = x^2$.

2. Let $S(n) = 1^2 + 2^2 + \ldots + n^2$. Note that $\Delta S(n) = S(n+1) - S(n) = (n+1)^2$ is a quadratic polynomial.

   (a) Verify that $\Delta^4 S(n) = 0$. This indicates that $S(n)$ is a polynomial of degree 3.

   (b) Find $S(n)$ by evaluating it at four points $n = 1, 2, 3, 4$ and interpolating a cubic polynomial through them.

3. Neville's method can be implemented with the following algorithm

NEVILLE($x, x_1, x_2, \ldots, x_n, f_1, f_2, \ldots, f_n$)

for $i \leftarrow 1, 2, \ldots, n$
do
  $N_{i,1} \leftarrow f_i$
  $d_i \leftarrow x - x_i$
for $k \leftarrow 2, 3, \ldots, n$
do
  for $i \leftarrow k, k+1, \ldots, n$
  do
    $N_{i,k} \leftarrow N_{i,k-1} + d_i \left(N_{i,k-1} - N_{i-1,k-1}\right) / (x_i - x_{i-k+1})$
return $N_{n,n}$

   (a) How many floating-point operations (additions, subtractions, multiplications, and divisions) does this algorithm require?

   (b) Modify the algorithm so that only one column of the matrix N is stored in memory instead of the whole matrix.

4. (Programming) In this assignment, we revisit nonlinear equations. One method of solving nonlinear equations $f(x) = 0$ is *inverse interpolation*. Taking several initial guesses for the root $c_0, c_1, \ldots, c_n$ and the corresponding function values $f_0, f_1, \ldots, f_n$, the inverse function $x(f)$ is interpolated (i.e., with a polynomial of degree n) and then evaluated at $f = 0$. This produces the next iteration

$$c_{n+1} = P_n(0)\,; \tag{2}$$
$$f_{n+1} = f(c_{n+1})\,. \tag{3}$$

   (a) Show that the method of inverse interpolation with $n = 1$ is equivalent to the secant method.

   (b) Implement both the secant method and the method of quadratic inverse interpolation ($n = 2$).

   (c) Use your programs to solve the equation

$$f(x) = x + e^x = 0\,. \tag{4}$$

   Use the initial guesses $c_0 = -1$ and $c_1 = 0$ in the secant method. Use the initial guesses $c_0 = -2$, $c_1 = -1$, and $c_2 = 0$ in the inverse quadratic interpolation. Perform the computations with the double precision and output tables of the form

n	c_n	$\lvert f_n \rvert$

   Which of the two methods takes less iterations to find the root with the tolerance

$$\lvert f_n \rvert < 10^{-15}\ ?$$

   Which method takes less operations?

5. (Programming) The following table contains the number of medals won by the United States at the winter Olympic games:

Year	Location	Gold	Silver	Bronze	Total	Points
1924	CHAMONIX	1	2	1	4	8
1928	SAINT MORITZ	3	2	2	7	15
1932	LAKE PLACID	6	4	2	12	28
1936	GARMISH PARTENKIRCHEN	1	0	3	4	6
1948	SAINT MORITZ	3	5	2	10	21
1952	OSLO	4	6	1	11	25
1956	CORTINA D'AMPEZZO	2	3	2	7	14
1960	SQUAW VALLEY	3	4	3	10	20
1964	INNSBRUCK	1	2	4	7	11
1968	GRENOBLE	1	4	1	6	12
1972	SAPPORO	3	2	3	8	16
1976	INNSBRUCK	3	3	4	10	19
1980	LAKE PLACID	6	4	2	12	28
1984	SARAJEVO	4	4	0	8	20
1988	CALGARY	2	1	3	6	11
1992	ALBERTVILLE	5	4	2	11	25
1994	LILLEHAMMER	6	5	2	13	30
1998	NAGANO	6	3	4	13	28

The points are computed with the formula

$$\text{Points} = 3 \times \text{Gold} + 2 \times \text{Silver} + \text{Bronze}\ .$$

(a) Implement the Neville interpolation algorithm.

(b) Apply it to predict the number of points that US would have won if the Olympic games took place in 1944. First, use the points from all the years in the table to do the interpolation. Next, use only four values from the years 1932–1952. Output the Neville table (the matrix $N_{i,j}$) for the second case.

(c) Apply your program to predict the number of points that US will get in 2002. First, use the points from all the years in the table. Next, use only the last five values from the years 1984-1998. Output the Neville table for the second case. Do you think this is a reliable prediction? Explain.