NUMERICAL ANALYSIS
PROGRAM SET C

PAUL L. BAILEY

Abstract. Create the following programs using Visual C++.

Program 1. Create a library of functions to manipulate polynomials, including subroutines to perform the following operations:

- **(a)** Return the zero polynomial.
- **(b)** Return the one polynomial.
- **(c)** Multiply a polynomial by a constant.
- **(d)** Add two polynomials.
- **(e)** Multiply two polynomials.
- **(f)** Return the derivative of a polynomial.
- **(g)** Return the integral of a polynomial (with zero as the constant term).
- **(h)** Print a polynomial.
- **(i)** Evaluate a polynomial.

Program 2. Create a function to compute the interpolation polynomial of a table using the Lagrange method. This should use a subroutine to compute the cardinal functions from a table, then combine them using the polynomial functions from Program 1.

Program 3. Create a function to compute the interpolation polynomial of a table using the Newton method. This should use a subroutine to first compute the Newton coefficients of the table (which are different from the coefficients of the polynomial), then prepare the standard form of the polynomial (with the standard coefficients) using the polynomial functions from Program 1.

The following program uses these types:

```
#define MAX 256
typedef double (*FNP)(double);
typedef double ARR[MAX][MAX];
```

Program 4. Create a function to produce Richardson extrapolation derivative estimates.
Syntax: `void rich(FNP f,double x,int n,double h,ARR D)`
where `rich` is the name of the function, `x` is the point at which we wish to estimate the derivative, `n` is the number of Richardson extrapolation rows, `h` is the initial Δx, and `D` is the return array.

Department of Mathematics and CSCI, Southern Arkansas University
E-mail address: plbailey@saumag.edu

Date: October 15, 2003.