Answers to Assignment #7

1. We must have

$$s_2(0) = s_1(0) = 1 + c; \quad s_2'(0) = s_1'(0) = 3c;$$
$$s_2''(0) = s_1''(0) = 6c; \quad s_2''(1) = 0.$$

This suggests

$$s_2(x) = 1 + c + 3cx + \frac{6c}{2!}x^2 + Ax^3.$$

The last condidion $[s_2''(1) = 0]$ gives $6c + 6A = 0$, hence $A = -c$. Thus,

$$s_2(x) = 1 + c + 3cx + 3cx^2 - cx^3.$$

The requirement $s(1) = -1$ is equivalent to $s_2(1) = -1$, that is,

$$1 + c + 3c + 3c - c = 1 + 6c = -1, \quad c = -\frac{1}{3}.$$

2. (a) SUBROUTINE (for tridiagonal system)

```
      subroutine tri(n,a,b,c,u,v)
c
c Gauss elimination without pivoting for a (nonsymmetric)
c tridiagonal system with diagonal elements a, lower
c subdiagonal elements b, upper superdiagonal elements c,
c and right-hand vector v. The solution vector is u.
c
      dimension a(n),b(n),c(n),u(n),v(n)
      do 10 i=2,n
        f=b(i-1)/a(i-1)
        a(i)=a(i)-f*c(i-1)
        v(i)=v(i)-f*v(i-1)
   10 continue
      u(n)=v(n)/a(n)
      do 20 i=n-1,1,-1
        u(i)=(v(i)-c(i)*u(i+1))/a(i)
   20 continue
      return
      end
```

(b) PROGRAM (for natural spline)

```
c
c Natural cubic spline interpolant to function fnc for the
c partition x(1)<x(2)<...<x(n).
c
      dimension xx(11),f(11),dx(10),df(10),a(11),b(11),c(11),
     *u(11),v(11),c0(10),c1(10),c2(10),c3(10),errm(10)
      write(*,1)
    1 format(/)
      n=11
      fnm1=float(n-1)
      np=51
      fnpm1=float(np-1)
      do 10 i=1,n
```

```
        xx(i)=float(i-1)/fnm1
c       xx(i)=(float(i-1)/fnm1)**2
        f(i)=fcn(xx(i))
   10 continue
      do 20 i=1,n-1
        dx(i)=xx(i+1)-xx(i)
        df(i)=(f(i+1)-f(i))/dx(i)
   20 continue
      a(1)=2.
      c(1)=1.
      b(n-1)=1.
      a(n)=2.
      v(1)=3.*df(1)
      v(n)=3.*df(n-1)
      do 30 i=2,n-1
        b(i-1)=dx(i)
        a(i)=2.*(dx(i-1)+dx(i))
        c(i)=dx(i-1)
        v(i)=3.*(dx(i)*df(i-1)+dx(i-1)*df(i))
   30 continue
      call tri(n,a,b,c,u,v)
      do 40 i=1,n-1
        c0(i)=f(i)
        c1(i)=u(i)
        c3(i)=(u(i+1)+u(i)-2.*df(i))/(dx(i)**2)
        c2(i)=(df(i)-u(i))/dx(i)-c3(i)*dx(i)
   40 continue
      do 60 i=1,n-1
        errm(i)=0.
        do 50 j=1,np
          x=xx(i)+float(j-1)*dx(i)/fnpm1
          t=x-xx(i)
          s=c3(i)
          s=t*s+c2(i)
          s=t*s+c1(i)
          s=t*s+c0(i)
          y=fcn(x)
```

```
          err=abs(s-y)
          if(err.gt.errm(i)) errm(i)=err
   50   continue
        write(*,2) i,errm(i)
    2   format(1x,i2,e13.4)
   60 continue
      stop
      end

      function fcn(x)
      fcn=exp(-x)
c     fcn=sqrt(x)**5
      return
      end
```

(c) PROGRAM (for complete spline)

```
c
c Complete cubic spline interpolant to function fnc for the
c partition x(1)<x(2)<...<x(n).
c
      dimension xx(11),f(11),dx(10),df(10),a(11),b(11),c(11),
     *u(11),v(11),c0(10),c1(10),c2(10),c3(10),errm(10)
      write(*,1)
    1 format(/)
      n=11
      fnm1=float(n-1)
      np=51
      fnpm1=float(np-1)
      do 10 i=1,n
        xx(i)=float(i-1)/fnm1
c       xx(i)=(float(i-1)/fnm1)**2
        f(i)=fcn(xx(i))
   10 continue
c
c The next two statements are new and do not occur in the
c preceding program of (b)
```

```
c
      fder1=fcnd(0.)
      fdern=fcnd(1.)
      do 20 i=1,n-1
        dx(i)=xx(i+1)-xx(i)
        df(i)=(f(i+1)-f(i))/dx(i)
   20 continue
c
c The next six statements differ from the corresponding statements
c in the preceding program of (b)
c
      a(1)=1.
      c(1)=0.
      b(n-1)=0.
      a(n)=1.
      v(1)=fder1
      v(n)=fdern
      do 30 i=2,n-1
        b(i-1)=dx(i)
        a(i)=2.*(dx(i-1)+dx(i))
        c(i)=dx(i-1)
        v(i)=3.*(dx(i)*df(i-1)+dx(i-1)*df(i))
   30 continue
      call tri(n,a,b,c,u,v)
      do 40 i=1,n-1
        c0(i)=f(i)
        c1(i)=u(i)
        c3(i)=(u(i+1)+u(i)-2.*df(i))/(dx(i)**2)
        c2(i)=(df(i)-u(i))/dx(i)-c3(i)*dx(i)
   40 continue
      do 60 i=1,n-1
        errm(i)=0.
        do 50 j=1,np
          x=xx(i)+float(j-1)*dx(i)/fnpm1
          t=x-xx(i)
          s=c3(i)
          s=t*s+c2(i)
```

```
          s=t*s+c1(i)
          s=t*s+c0(i)
          y=fcn(x)
          err=abs(s-y)
          if(err.gt.errm(i)) errm(i)=err
   50   continue
        write(*,2) i,errm(i)
    2   format(1x,i2,e13.4)
   60 continue
      stop
      end

      function fcn(x)
      fcn=exp(-x)
c     fcn=sqrt(x)**5
      return
      end

      function fcnd(x)
      fcnd=-exp(-x)
c     fcnd=2.5*sqrt(x)**3
      return
      end
```

OUTPUT

```
  i   err max(i)

  1   0.4903E-03   f(x)=exp(-x)
  2   0.1316E-03   natural spline
  3   0.3505E-04   uniform partition
  4   0.9596E-05
  5   0.2205E-05
  6   0.4172E-06
  7   0.3457E-05
  8   0.1281E-04
  9   0.4846E-04
```

```
10   0.1804E-03

 1   0.2980E-06   f(x)=exp(-x)
 2   0.2384E-06   complete spline
 3   0.2384E-06   uniform partition
 4   0.2384E-06
 5   0.1788E-06
 6   0.1788E-06
 7   0.1788E-06
 8   0.1192E-06
 9   0.1192E-06
10   0.1192E-06

 1   0.2052E-03   f(x)=x**(5/2)
 2   0.5339E-04   natural spline
 3   0.1619E-04   uniform partition
 4   0.2753E-05
 5   0.1311E-05
 6   0.9805E-05
 7   0.3487E-04
 8   0.1326E-03
 9   0.4933E-03
10   0.1841E-02

 1   0.4435E-04   f(x)=x**(5/2)
 2   0.1000E-04   complete spline
 3   0.4714E-05   uniform partition
 4   0.3949E-06
 5   0.9984E-06
 6   0.5662E-06
 7   0.5662E-06
 8   0.3874E-06
 9   0.2980E-06
10   0.3576E-06

 1   0.6690E-06   f(x)=x**(5/2)
 2   0.2355E-06   natural spline
```

```
 3   0.1175E-05   nonuniform partition
 4   0.1695E-05
 5   0.1585E-05
 6   0.1644E-04
 7   0.6802E-04
 8   0.3295E-03
 9   0.1475E-02
10   0.6545E-02

 1   0.1081E-06   f(x)=x**(5/2)
 2   0.6055E-06   complete spline
 3   0.9125E-06   nonuniform partition
 4   0.1356E-05
 5   0.1727E-05
 6   0.2153E-05
 7   0.2533E-05
 8   0.2906E-05
 9   0.3368E-05
10   0.3695E-05
```

COMMENTS

- Testing: For the natural spline, the error should be exactly zero if f is any linear function. (Not for arbitrary cubics, since f'' does not vanish at $x = 0$ and $x = 1$, unless f is linear.) For the complete spline, the error is zero for any cubic, if one sets $m_1 = f'(0)$ and $m_n = f'(1)$. Example: $f(x) = x^3,\ m_1 = 0,\ m_n = 3$.
- The natural spline for the uniform partition is relatively inaccurate near the endpoints, as expected.
- The complete spline is uniformly accurate for $f(x) = e^{-x}$ but still relatively inaccurate near $x = 0$ for $f(x) = x^{5/2}$ on account of the "square root" singularity (of f''') at $x = 0$.
- Nonuniform partition (for $f(x) = x^{5/2}$): The natural spline is accurate near $x = 0$ because of the nodes being more dense there, but is still inaccurate at the other end. The complete spline is remarkably accurate at both ends, as well as elsewhere.