Appendix C

Lawrence Livermore Loops

The following Fortran loops have become a standard benchmark for measuring performance in a scientific computing environment.

1.
```
      DO  1  K = 1,400
1        X(K) = Q + Y(K)*(R*Z(K+10) + T*Z(K+11))
```

2.
```
      DO  2  K = 1,996,5
2        Q = Q + Z(K)*X(K) + Z(K+1)*X(K+1) + Z(K+2)*X(K+2)
     1      + Z(K+3)*X(K+3) + Z(K+4)*X(K+4)
```

3.
```
      DO  3  K = 1,1000
3        Q = Q + Z(K)*X(K)
```

4.
```
      DO  4  J = 30,870,5
         X(L-1) = X(L-1) - X(LW)*Y(J)
4        LW = LW + 1
```

5.
```
      DO  5  I = 2,998,3
         X(I) = Z(I)*(Y(I) - X(I-1))
         X(I+1) = Z(I+1)*(Y(I+1) - X(I))
5        X(I+2) = Z(I+2)*(Y(I+2) - X(I+1))
```

6.
```
      DO  6  J = 3,999,3
         I = 1000 - J + 3
         X(I) = X(I) - Z(I)*X(I+1)
         X(I-1) = X(I-1) - Z(I-1)*X(I)
6        X(I-2) = X(I-2) - Z(I-2)*X(I-1)
```

7.
```
      DO  7  M = 1,120
         X(M) = U(M) + R*(Z(M) + R*Y(M))
     1              + T*(U(M+3) + R*(U(M+2) + R*U(M+1))
     2              + T*(U(M+6) + R*(U(M+5) + R*U(M+4)))
7     CONTINUE
```

8.
```
      DO  8  KX = 2,3
         DO  8  KY = 2, 21
            DU1 = U1(KX,KY+1,NL1) - U1(KX,KY-1,NL1)
            DU2 = U2(KX,KY+1,NL1) - U2(KX,KY-1,NL1)
            DU3 = U3(IX,KY+1,NL1) - U3(KX,KY-1,NL1)
            U1(KX,KY,NL2) = U1(KX,KY,NL1) + A11*DU1 + A12*DU2
     1           + A13*DU3 + SIG*(U1(KX+1,KY,NL1)
     2           - 2*U1(KX,KY,NL1) + U1(KX-1,KY,NL1))
            U2(KX,KY,NL2) = U2(KX,KY,NL1) + A21*DU1 + A22*DU2
     1           + A23*DU3 + SIG*(U2(KX+1,KY,NL1)
     2           - 2*U2(KX,KY,NL1) + U2(KX-1,KY,NL1))
            U3(KX,KY,NL2) = U3(KX,KY,NL1) + A31*DU1 + A32*DU2
     1           + A33*DU3 + SIG*(U3(KX+1,KY,NL1)
     2           - 2*U3(KX,KY,NL1) + U3(KX-1,KY,NL1))
   8     CONTINUE
```

9.
```
      DO  9  I = 1,100
         PX(1,I) = BM28*PX(13,I) + BM27*PX(12,I) + BM26*PX(11,I)
     1           + BM25*PX(10,I) + BM24*PX(9,I) + BM22*PX(7,I)
     2           + CO*(PX(5,I) + PX(6,I)) + PX(3,I)
   9     CONTINUE
```

10.
```
      DO  10  I = 1,100
         AR = CX(5,I)
         BR = AR - PX(5,I)
         PX(5,I) = AR
         CR = BR - PX(6,I)
         PX(6,I) = BR
         AR = CR - PX(7,I)
         PX(7,I) = CR
         BR = AR - PX(8,I)
         PX(8,I) = AR
         CR = BR - PX(9,I)
         PX(9,I) = BR
         AR = CR - PX(10,I)
         PX(10,I) = CR
         BR = AR - PX(11,I)
         PX(11,I) = AR
         CR = BR - PX(12,I)
         PX(12,I) = BR
         PX(14,I) = CR - PX(13,I)
         PX(13,I) = CR
  10     CONTINUE
```

11.
```
      X(1) = Y(1)
      DO  11  K = 2,1000
11       X(K) = X(K-1) + Y(K)
```

12.
```
      DO  12  K = 1,999
12       X(K) = Y(K+1) - Y(K)
```

13.
```
      DO  13  IP = 1,128
         I1 = P(1,IP)
         J1 = P(2,IP)
         P(3,IP) = P(3,IP) + B(I1,J1)
         P(4,IP) = P(4,IP) + C(I1,J1)
         P(1,IP) = P(1,IP) + P(3,IP)
         P(2,IP) = P(2,IP) + P(4,IP)
         I2 = P(1,IP)
         J2 = P(2,IP)
         P(1,IP) = P(1,IP) + Y(I2+32)
         P(2,IP) = P(2,IP) + Z(J2+32)
         I2 = I2 + E(I2+32)
         J2 = J2 + F(J2+32)
         H(I2,J2) = H(I2,J2) + 1.0
13    CONTINUE
```

14.
```
      DO  14  K = 1,150
         IX = GRD(K)
         XI = IX
         VX(K) = VX(K) + EX(IX) + (XX(K) - XI)*DEX(IX)
         XX(K) = XX(K) + VX(K) + FLX
         IR = XX(K)
         RI = IR
         RXI = XX(K) - RI
         IR = IR - (IR/64)*64
         XX(K) = RI + RXI
         RH(IR) = RH(IR) + 1.0 - RXI
         RH(IR+1) = RH(IR+1) + RXI
14    CONTINUE
```