Curve Fitting

The table of contents leads us to `POLFIT` to fit a function linear in its parameters. The routine `PCOEF` allows us to express the fit obtained via `POLFIT`.

Note that for non-linear fitting, the routine `SNLS1` should be used.

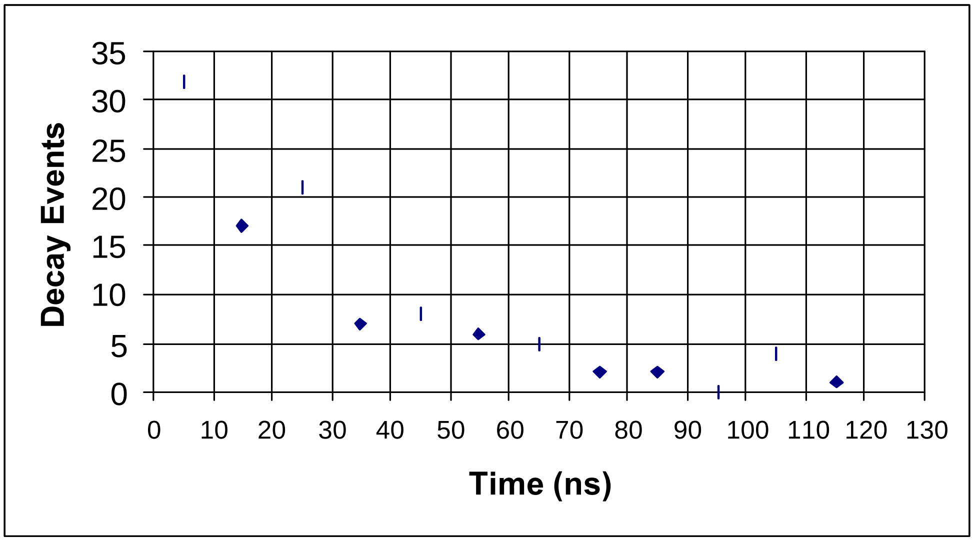
As an example, let us fit an experimental observation to theory: The experiment measures the number of pi mesons that decay in intervals of 10 ns. Theory predicts that if N(t) is the #of pions at time t, then dN/dt = -(N_0/ô)$e^{-t/ô}$

```
program piDecay
Implicit none
real*8 x(12), w(12)
real*8 y(12) /32, 17, 21, 7, 8, 6, 5, 2, 2, 0.1, 4,
1/
real*8 A(100), R(12), eps, TC(12), yy
integer*4 i, maxDeg, nDeg, j, n, status
maxDeg = 1
n = 12
eps = 0.
do i=1, n
   x(i) = i* 10 - 5
   y(i) = log(y(i))
end do
w(1) = -1
call dpolft(n, x, y, w, maxDeg, nDeg, eps, R, status, A)
print*, "Status  = ", status
print*, "nDegree = ", nDeg
print*, "EPS     = ", eps
call dpcoef(nDeg, 0., TC, A)
print*, "Intercept / Slope: ", TC(1), TC(2)
print*, "pi meson lifetime (26 ns) = ", -1./TC(2)
end
```

Running the above program yields:

```
Status  =  1
nDegree =  1
EPS     =   0.886860617
Intercept / Slope:   3.51737089 -0.0346783699
pi meson lifetime (26 ns) =   28.8364189
```

TIME	E
5	32
15	17
25	21
35	7
45	8
55	6
65	5
75	2
85	2
95	0.1
105	4
115	1



TIME	E	FIT
5	32	28.33164
15	17	20.02928
25	21	14.15987
35	7	10.01043
45	8	7.076959
55	6	5.003114
65	5	3.536993
75	2	2.500506
85	2	1.767753
95	0.1	1.249728
105	4	0.883505
115	1	0.624601