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 number of pions at time \( t \), then
\[
\frac{dN}{dt} = -\left( \frac{N_0}{\alpha} \right) e^{-t/\alpha}
\]

```fortran
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