Tutorials and worked examples for simulation, curve fitting, statistical analysis, and plotting.
http://www.simfit.org.uk

Given a model defining several equations in one or more variables, the integrals can be estimated over a hyper-rectangular region defined by fixed limits.

The following procedure is required for $n \geq 1$ functions of $m \geq 1$ variables.

1. Create a file defining the n functions of m variables to be integrated.
2. Open program **usermod** and input the file defining n function of m variables.
3. It is necessary to explicitly indicate that n functions of m variables are required and the values for n and m must be specified correctly.
4. Program **usermod** then checks that the function is defined correctly.
5. The range of integration required must be defined by editing the vectors $BLIM$ and $TLIM$ to specify the m lower and upper limits for the corresponding variables.
6. The absolute error $EPSABS$ and relative error $EPSREL$ parameters required must be set.
7. Integration can then be requested but the result should only accepted if $IFAIL = 0$ on completion.
8. If $IFAIL = 1$ on exit, then re-entry for continued iterations will be offered, otherwise some of the previous parameters will have to be adjusted and the integration repeated.

From the main SimFIT menu, choose [A/Z], open program **usermod**, then read in test file `d01eaf_e.mod` which defines the the integrand used to evaluate the following integral The program accepts a user defined model for n functions of m variables and estimates the n integrals

$$I_i = \int_{A_1}^{B_1} \int_{A_2}^{B_2} \ldots \int_{A_m}^{B_m} f_i(x_1, x_2, \ldots, x_m)\, dx_m \ \ldots \ dx_2\, dx_1$$

for $i = 1, 2, \ldots, n$, where the limits are taken from the arrays $A_i = blim(i)$ and $B_i = tlim(i)$. The procedure only returns IFAIL = 0 when

$$\max_i(ABSEST(i)) \leq \max(EPSABS, EPSREL \times \max_i |FINEST(i)|),$$

where $ABSEST(i)$ is the estimated absolute error in $FINEST(i)$, the final estimate for integral i, as described for NAG routine D01EAF.

The n functions defined by SimFIT test file `d01eaf_e.mod` are

$$f_j = \log(x_1 + 2x_2 + 3x_3 + 4x_4)\sin(j + x_1 + 2x_2 + 3x_3 + 4x_4) \text{ for } j = 1, 2, \ldots, 10$$

while the results from integration are listed in the following tables.

Results from the integration of `d01eaf_e.mod`

$IFAIL$	0 (from D01EAF)
$EPSABS$	1.000E-06
$EPSREL$	1.000E-03
$MINCLS$	459 (Function evaluations)
$TESTER$	4.417E-04 (Error threshold: * where exceeded)

Variable	*BLIM*	*TLIM*
1	0.0	1.0
2	0.0	1.0
3	0.0	1.0
4	0.0	1.0

Function	*INTEGRAL*	*ABSEST*
1	3.8352146E-02	1.8779E-04
2	4.0118447E-01	2.3766E-04
3	3.9516964E-01	1.6379E-04
4	2.5837668E-02	1.7314E-04
5	-3.6724934E-01	2.3574E-04
6	-4.2268900E-01	1.5493E-04
7	-8.9510341E-02	1.5503E-04
8	3.2596371E-01	2.2910E-04
9	4.4174823E-01	4.5854E-03 *
10	1.5139146E-01	5.1370E-04 *

The other parameters in these tables that have not already been defined have the following meanings.

MINCLS	Number of calls to the subroutine for function evaluations.
TESTER	Maximum error estimate acceptable so that items larger than this (if any) are indicated by the symbol * in the listing (as for functions 9 and 10). There can be a few * symbols and still have IFAIL = 0 on exit as a slightly weaker test than this is performed by the numerical integrator.
INTEGRAL	Integral for listed function.
ABSEST	Error estimate for listed function.

The SIMFIT test file defining these 10 functions of 4 variables is now listed.

```
%
...
model for the 10 functions in 4 variables required to demonstrate D01EAF
f_j = log(x_1 + 2x_2 + 3x_3 + 4x_4)(sin(j + x_1 + 2x_2 + 3x_3 + 4x_4)
      for j = 1, 2, ..., 10
...
%
10 equations
4 variables
0 parameters
%
begin{expression}
A = y(1) + 2y(2) + 3y(3) + 4y(4)
B = log(A)
 f(1) = B*sin(1 + A)
 f(2) = B*sin(2 + A)
 f(3) = B*sin(3 + A)
 f(4) = B*sin(4 + A)
 f(5) = B*sin(5 + A)
 f(6) = B*sin(6 + A)
```

```
 f(7) = B*sin(7 + A)
 f(8) = B*sin(8 + A)
 f(9) = B*sin(9 + A)
f(10) = B*sin(10 + A)
end{expression}
%
```

Note the use of dummy variables A and B to avoid re-calculations.