ADMB

Automatic Differentiation Model Builder

ADModel builder

Is a complete fourth generation language (4GL) programming environment, based on automatic differentiation and specialized for the development of non-linear optimization applications.

ADModel Builder was developed by David Fournier in the early 1990s and is rooted in one of the original C++ implementations of reverse mode automatic differentiation, Fournier's AUTODIF library.

Core components

- Template with keywords and special data types to facilitate and simplify model development.
- Parser which generates standard C++ from the 4GL template.
- Overloading of *all* C++ operators and math library functions
- Specialized functions and class libraries for building statistical models and reverse-mode automatic differentiation
- Suite of efficient numerical function minimizers to exploit high precision derivatives
- Class libraries for vector and matrix operations
- Flexible multidimensional container classes

Features

- Fast, efficient, and robust optimization
- Automatic estimation of the variance of all estimated model parameters and optional estimation of the variance of any model variable.
- Random effects models
- Metropolis-Hastings algorithm for Markov Chain Monte Carlo integration
- Higher order derivatives by a combination of forward and reverse mode
- Likelihood profile estimation

The ADMB Project

- Create free, open-source software community
    Libraries and source code available for public download
- Promote wider use of ADMB
    Training workshops
    Tools and tips posted on website and user group
    Student awards
- Extend ADMB capabilities
    Broad suite of computing platforms
    Optimizations for speed
    Multithreading

http://admb-project.org/

Applications

ADMB provides fast, efficient and robust numerical parameter estimation using automatic differentiation, fully integrated class libraries and specialized functions. ADMB is suitable for both data-rich and parameter-rich applications and for applications that require extensive optimization.

Highly-parameterized nonlinear models. Fisheries stock assessment models with thousands of parameters (e.g. MULTIFAN-CL, Hampton and Fournier 2001; A-SCALA, Maunder and Watters 2003; Stock Synthesis II, Methot, 2005).

General models for fisheries stock assessment (Coleraine, Hilborn et al. 2000, http://www.fish.washington.edu/research/coleraine/; Stock Synthesis II, Methot, 2005) and marine mammal assessment (Maunder and Hilborn 2000; Breen et al. 2003).

Integrating several different data types. Hampton and Fournier (2001) and Maunder (2001a) integrate mark-recapture data with catch-at-length and catch-at-age data, respectively.

Nonlinear models with large data sets. Maunder (2001b) combined an age-structure model with a general linear model (GLM) and fit the model to 24,183 data points in the GLM component (standardizing catch-and-effort data) and to catch-at-age data.

Management strategy evaluation. Simulation analysis of management strategies for which the management strategy includes the definition of the stock assessment procedure (Parma 2002) and requires that a stock assessment model is fit to the simulated data for each year of the simulation.

Random effects modelling. Automatic evaluation of integrated likelihood via the Laplace approximation, (Skaug and Fournier, 2006; Kitakado et al., 2006; Cope and Punt, 2007; Varin et al, 2005).

Analysis of animal behavior. Estimation of thermoregulatory abilities of freely swimming bigeye tuna (Holland et al, 1992). Estimation of movement, mortality and residence times from conventional tagging experiments (Sibert et al, 1999; Adam et al, 2005). Estimation of tracks from marine animals tagged with electronic devices, (Sibert, et al, 2003; Nielsen et al, 2006; Nielsen and Sibert, 2007; Lam et al, 2008).

Ecosystem and global climate change models. Senina et al (2008).

Global user base

Applied to management and conservation of more than 150 different sensitive endangered species and commercially valuable fish stocks around the world.

Institutions where ADMB is used. http://admb-project.org/user-base.html
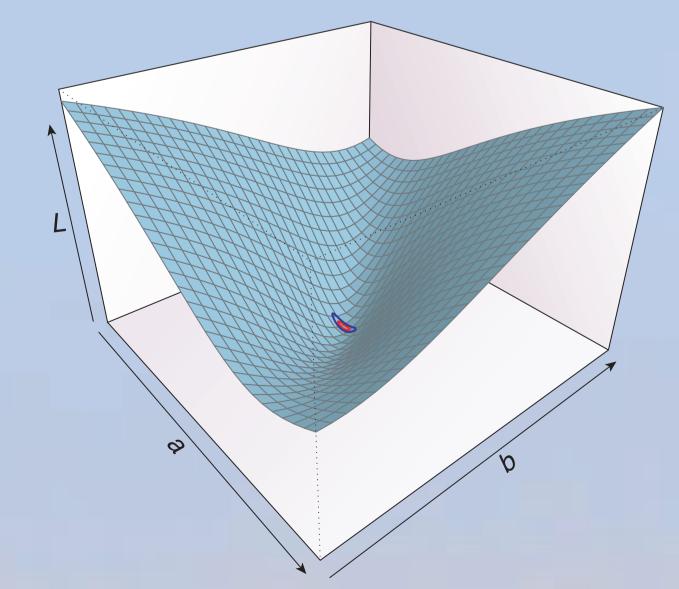
4GL Template file:

```
DATA_SECTION
  init_int nobs
  init_vector Y(1,nobs)
  init_vector x(1,nobs)
PARAMETER_SECTION
  init_number a
  init_number b
  vector pred_Y(1,nobs)
  objective_function_value f
PROCEDURE_SECTION
  pred_Y=a*x+b;
  f=sum(square(pred_Y-Y));
  f=nobs/2.*log(f);
```

Simple ADMB template file for linear regression. Note the specialized classes for statistical modeling and the use of overloaded vector operators to avoid loops.

Data file:

```
# number of observations 10
# observed Y values
1.4  4.7  5.1  8.3  9.0  14.5
14.0  13.4  19.2  18
# observed x values
-1 0 1 2 3 4 5 6 7 8
```

The data file corresponding to the above template. The data appear in the same order as the variables are declared in the template.

Output file:

```
index  name     value      std dev      1       2
 1      a     1.9091e+00  1.5547e-01   1.0000
 2      b     4.0782e+00  7.0394e-01  -0.7730  1.0000
```

For each estimated model parameter, ie each variable of type init_number in the PARAMETER_SECTION of the template file, this file lists the estimate parameter value along with its standard deviation and correlation to other model parameters as estimated from the inverse Hessian matrix.

Value of the negative log likelihood (L) as a function of both model parameters (a and b). The colored ellipses near the minimum indicate the approximate 99% (blue) and 95% (red) joint confidence region around the minimum. The minimum is interpreted as the point estimates of a and b.