Summer School in Statistics for
Astronomers & Physicists
June 15–17, 2005

# Center for Astrostatistics

# Pennsylvania State University

Computational algorithms for astrostatistics:

Principles of Statistical Algorithms

Thriyambakam Krishnan
Systat Software Asia–Pacific Limited
Bangalore, India

# What is an algorithm?

A list of instructions for carrying out a job
It should be clear and unambiguous

Knuth (1997): Five properties of an algorithm:

1. finiteness—executed with finite resources

2. definiteness—instructions must be completely and unambiguously defined

3. should have an input

4. should have an output

5. effective—instructions should be executable

## Are statistical algorithms any different from other algorithms?

**Some mathematical (deterministic) algorithms in Statistics**:

1. solving equations or root-finding
2. optimization
3. matrix computations (inversions, decompositions, etc.)
4. search
5. permutations
6. function approximation
7. numerical quadrature

**Algorithms in Statistics involving randomness**:

1. Experimental designs
2. Design of a Survey
3. Random sampling
4. Randomization methods
5. Bootstrap and other resampling methods
6. Simulation methods
7. Rejection sampling, Adaptive Rejection Sampling
8. Markov chain Monte Carlo (MCMC)
9. Monte Carlo integration

# Concepts relating to algorithms:

1. **Accuracy**: absolute or relative error of a quantity approximating another quantity;

2. **Precision**: degree of agreement over repetitions; number of digits that are the same; number of (significant) digits reported (different from statistical notion of precision as smaller standard deviation)

3. **Efficiency**: measured in terms of such factors as storage requirements, execution speed, number of basic operations (computational complexity), maybe even complexity of program

4. **Numerical Stability**: Results of the same accuracy regardless of the data

5. **Problem Solvability**: An algorithm solves a problem only if it produces 'correct' solution for all instances of a problem

# **Ad hoc vs high-quality algorithms**:

End-user is interested in results not in the process;
so are ad hoc algorithms adequate?

## Points in favor of high-quality algorithms:

- improved use of our own time
  - adapt to new data;
  - adapt to change in analysis
- improved quality of analysis
  - analyst and audience to be clear & precise
  - communicability, verifiability, reproducibilty, scientific validity
- acceptance as correct implementation of valid computational methods
- software systems need to combine ease of use with good quality algorithms

## **How to Evaluate Algorithms**:

• Not as a black box, not just by its performance, but also by techniques used
• Criteria of accuracy, precision, efficiency, numerical stability and problem solvability
• Is it useful? Does it solve intended problems in general, in all cases? Does it solve nasty and pathological cases? Does it solve similar problems?
• Is it correct? In all cases? Guarantees of accuracy? Does it produce error and warning messages?
• How easy to implement? Storage? Portability? Computational complexity? Easy programmability?

## How to Specify an Algorithm:

- mathematical language
- programming language
- pseudo code
- computer-readable form

# Statistical Method vs Statistical Algorithm

Statistical formula or Statistical method vs
Statistical algorithms vs
Statistical computing

Formula vs algorithm; Procedure vs algorithm (Examples: Variance; Median; MLE)

Statistical theory $\Longrightarrow$ Statistical method $\Longrightarrow$ Statistical algorithm $\Longrightarrow$ Computer implementation

Example:

Theorems on MLE $\Longrightarrow$ maximization of log likelihood; information matrix and asymptotic covariance matrix $\Longrightarrow$ solution of likelihood equations and computation of information matrix $\Longrightarrow$ Implementation by Fisher's scoring method

# Algorithm vs Implementation

• algorithm vs computer implementation: algorithm is defined independent of its implementation (or program); implementation is a particular instantation with its own accuracy, etc.

• algorithms are designed and analyzed independent of implementation; proof under infinite precision of arithmetic

• **Different algorithms for same procedure or same formula: Examples below**

## **Characteristics of Statistical Algorithms**:

Algorithms for Statistical Inference and Decision-making need to compute a number of quantities

- Estimates with standard errors and/or confidence intervals
- Tests with p-values and power
- Classification with error rates
- Regression with analysis of variance and diagnostics

Algorithms need to be good for all these

## Example 1: Calculation of Standard Deviation

(Thisted, 1988, Altman et al., 2004):

**Usual Formula**: $s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$

**Algorithm A**: (a) sum; (b) mean; (c) deviation from mean; (d) squared deviation from mean; (e) sum of squared deviations from mean; (f) division by $(n-1)$.

**Algorithm B**: Alternative formula (Single-pass algorithm):

$$s^2 = \frac{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}{n(n-1)}$$

Sum and sum of squares in same pass; two multiplications, one squaring, one subtraction, one division. More efficient than A, less storage, less number of operations

**Example 1: Calculation of Standard Deviation: Continued**

**Algorithm C**: (a) Sort the $x_i$'s in ascending order; (b) compute sum; (c) compute mean; (d) compute deviations from mean; (e) compute squared deviations $z_i$'s from mean; (f) sort $z_i$'s in ascending order; (g) compute sum of $z_i$'s; (h) divide by $(n-1)$. 3-passes; greater complexity and less efficiency

**A** is numerically stable (avoids subtracting two large numbers); **B** is efficient; **C** is accurate

**Updating Algorithm**: new observation $x_{n+1}$

$$\bar{x}_{n+1} = \frac{1}{n+1}(n\bar{x}_n + x_{n+1})$$

$$s^2_{n+1} = \frac{n}{n+1}s^2_n + \frac{1}{n}(x_{n+1} - \bar{x}_{n+1})^2$$

## Example 2: Computation of Median:

Median $\mu_{1/2}$ minimizes $f(\mu) = E(|Y - \mu|)$

Sample observations $y_1, y_2, \ldots, y_n$

Sort: $y_{(1)}, y_{(2)}, \ldots, y_{(n)}$

if $n$ is odd n=2m+1, $y_{(m+1)}$ is median; else $\frac{y_{(m)}+y_{(m+1)}}{2}$ is median

**Median computing algorithm (without sorting)**:

Iteratively Reweighted Least Squares Algorithm to find sample median (without sorting):

If $\mu_{1/2}^{(k)}$ is the $k^{\text{th}}$ iterate, then

$$\mu_{1/2}^{(k+1)} = \frac{\sum_{i=1}^{k} w_i^{(k)} y_i}{\sum_{i=1}^{k} w_i^{(k)}},$$

where

$$w_i^{(k)} = |y_i - \mu_{1/2}^{(k)}|^{-1}$$

More efficient, since sorting is an inefficient procedure

**Example 3: Quantiles of Theoretical Distributions: Bisection Algorithm**:

$95^{\text{th}}$ percentile of F(3,7) with c.d.f. $F(x)$
Solution for $x$ of $G(x) = F(x) - 0.95 = 0$.
Assume we can compute $G(x)$.

Solution of equation by successive bisection
Start with two points $x_1, x_2$ such that

$$G(x_1) < 0, G(x_2) > 0$$

Bisect $x_1, x_2$, that is, compute $y = \frac{1}{2}(x_1 + x_2)$
Take $x_1, y$ or $y, x_2$ whichever satisfies
$G(x_1) < 0, G(y) > 0$ or $G(y) < 0, G(x_2) > 0$
Keep iterating until left and right are nearly the same.

## Example 3 Continued: F(3, 7)-distribution 95% Computations by Bisection:

4 5
4 4.5
4.25 4.5
4.25 4.375
4.315 4.375
4.345 4.375
4.345 4.360
4.345 4.352
4.3450 4.3485
4.3450 4.3472
4.3461 4.3472
4.34665

## **Example 4: A Permutation Test**:

Paired Comparison Test:

| Pre-treatment (A) | Post-Treatment (B) |
|---|---|
| 149 | 0 |
| 0 | 51 |
| 0 | 0 |
| 259 | 385 |
| 106 | 0 |
| 255 | 235 |
| 0 | 0 |
| 52 | 0 |
| 340 | 48 |
| 0 | 65 |
| 180 | 77 |
| 0 | 0 |
| 84 | 0 |
| 89 | 0 |
| 212 | 53 |
| 554 | 150 |
| 500 | 0 |
| 424 | 165 |
| 112 | 98 |
| 2600 | 0 |

Paired t-test vs Two-Sample t-test
Classical methods—assumption of normality, etc.
Not quite valid in this case

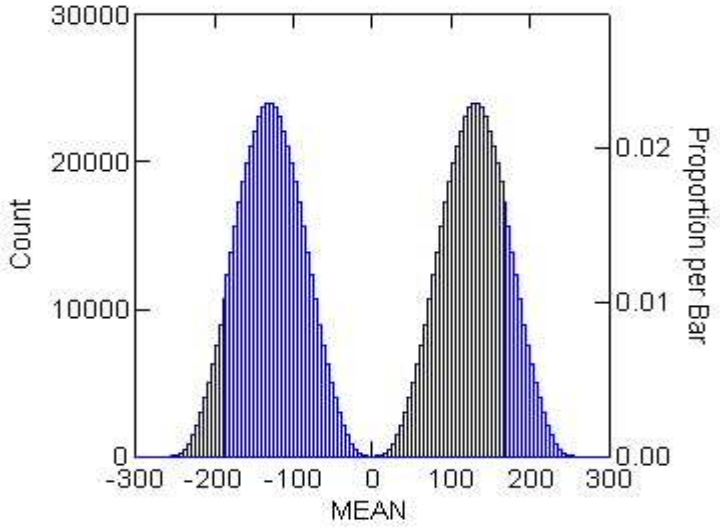
Paired samples t test on A vs B with 20 cases

```
Mean A              =    295.800000
Mean B              =     66.350000
Mean Difference     =    229.450000
SD Difference       =    579.606080
95.00% CI           = -41.813997 to  5.007E+02
          t                 =     1.770395
    df =     19     Prob =     0.092706
```

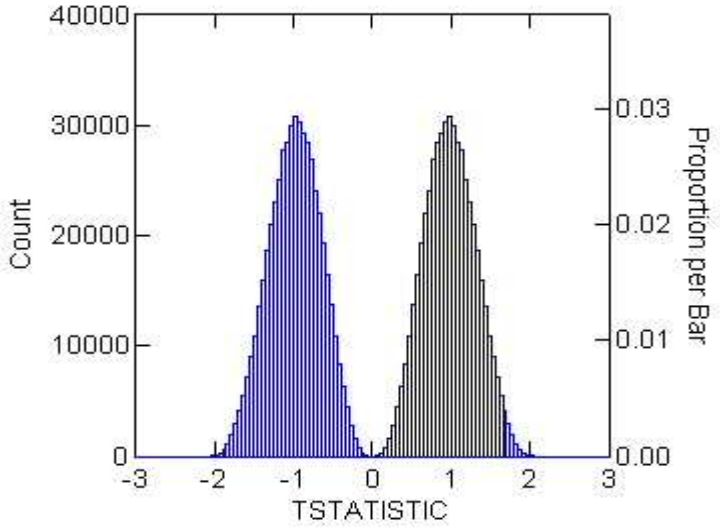
PERMUTATION TEST:

$2^{20}$ = over 1 million permutations (within each pair)
6544 observations in 'tstatistic' which are outside of (-1.770395,1.770395)
Proportion of observations in 'tstatistic' which are outside of (-1.770395,1.770395) is 0.006241

Permutation-based Mean Difference




Permutation-based $t$-Statistics



# Maximum likelihood estimation (MLE): An example

Observed data vector of frequencies out of a total of $n = 197$

$$\boldsymbol{y} = (y_1, y_2, y_3, y_4)^T$$
$$= (125, 18, 20, 34)^T,$$

from multinomial

$$\tfrac{1}{2} + \tfrac{1}{4}\theta,\ \tfrac{1}{4}(1-\theta),\ \tfrac{1}{4}(1-\theta),\ \text{and}\ \tfrac{1}{4}\theta$$

with $0 \leq \theta \leq 1$

Likelihood function $L(\boldsymbol{\theta})$:

Maximize $\ell(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta})$;

log is an increasing function

$\log L$ and derivatives are additive over independent observations

Score Function: Derivative $\dot{\ell}(\boldsymbol{\theta})$

MLE: $\widehat{\boldsymbol{\theta}}$: zero of the score function

provided second derivative $\ddot{\ell}$ is negative at solution

Score statistic: $\dot{\ell}(\boldsymbol{\theta})$: for fixed $\boldsymbol{\theta}$ as a random variable

$\ddot{\ell}(\widehat{\boldsymbol{\theta}})$: variance (covariance matrix) of $\widehat{\boldsymbol{\theta}}$.

$$\log L(\theta) = y_1 \log(2+\theta) + (y_2+y_3)\log(1-\theta) + y_4 \log\theta$$

$$\partial \log L(\theta)/\partial\theta = \frac{y_1}{2+\theta} - \frac{y_2+y_3}{1-\theta} + \frac{y_4}{\theta}$$

$$I(\theta;\, \boldsymbol{y}) = -\partial^2 \log L(\theta)/\partial\theta^2$$

$$= \frac{y_1}{(2+\theta)^2} + \frac{y_2+y_3}{(1-\theta)^2} + \frac{y_4}{\theta^2}$$

Likelihood equation is quadratic, can be solved directly

Newton-type methods;

Fisher's Scoring method:

uses $\mathcal{I}(\theta^{(k)})$ instead of $I(\theta^{(k)};\, \boldsymbol{y})$ on each iteration $k$

$$\mathcal{I}(\theta) = E_\theta\{I(\theta;\, \mathbf{Y})\}$$

$$= \frac{n}{4}\left\{\frac{1}{2+\theta} + \frac{2}{(1-\theta)} + \frac{1}{\theta}\right\}$$

**Newton-Raphson Method**: Iterative method starting from $\boldsymbol{\theta}^{(0)}$, update

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \dot{\ell}(\boldsymbol{\theta}^{(k)})\ddot{\ell}^{-1}(\boldsymbol{\theta}^{(k)})$$

until 'convergence' is reached

**Fisher's Scoring Method**: uses expected (Fisher) information $E(\ddot{\ell})$ instead of Newton-Raphson's observed information $\ddot{\ell}$.

• Scoring is a special statistically-tuned technique—produces parameter estimates as well as standard errors

• Success of method depends on starting values, iteration method, convergence criteria

• Newton-Raphson's quadratic convergence is useful; could be more sensitive to starting values

• EM algorithm is another alternative in some nasty problems; but extra efforts needed to produce standard errors

• In iterative algorithms of optimization, rates of convergence, sensitivity to initial values and ability to produce all statistics needed for a particular analysis are criteria

• Newton-type algorithms rely on quadratic approximations to $\log L$.

Newton's method is the gold standard for speed around $\widehat{\boldsymbol{\theta}}$

**Potential problems with Newton's method**:

1. expensive to evaluate observed information matrix;
2. far from $\widehat{\boldsymbol{\theta}}$, it can go up or down—not an ascent algorithm
3. positive definite modification needed—step-halving

**Approximation to observed information matrix**:

1. Steepest ascent: replace by $\boldsymbol{I}$
2. replace by expected information (Fisher's scoring)

$$J(\boldsymbol{\theta}) = E(\ddot{\ell}) = \text{Var}(\dot{\ell}(\boldsymbol{\theta})^T)$$

3.$J^{-1}(\widehat{\boldsymbol{\theta}})$): asymptotic covariance matrix of MLE

4. Current approximation $\boldsymbol{A}_k$ to observed information is updated by low-rank perturbation satisfying $-\boldsymbol{A}_{k+1}s_k g_k$, (called secant condition), where $s_k = \boldsymbol{\theta}_k - \boldsymbol{\theta}_{k+1}$, $g_k = (\dot{\boldsymbol{\theta}})_k^T - (\dot{\boldsymbol{\theta}})_{k+1}^T$. This is called **Quasi-Newton** method.

5. Observed information (Newton's) and expected information (Scoring) are asymptotically equivalent

• for local convergence Newton's is best
• computational complexity and numerical stability-wise EM is best
• scoring is in between—converges more quickly than EM and more stable than Newton's
• Quasi-Newton also similar behavior
• none of these variations is uniformly superior to the rest in terms of algorithm criteria for all data sets
• all these algorithms are in use—in books, in practice and in software packages
• practical strategy: hybrid algorithms: problem and data dependent: switch to another algorithm for a few iterations; very heuristic
• MLE by Newton, Fisher and its innumerable variants, and linear and nonlinear least-squares are the most often-used statistical algorithms

## Statistical algorithms dictated by computing limitations

• statistical algorithms tailored to circumvent computational limitations (Thisted, 1988): Examples: centroid methods of factor analysis; later supplanted by principal component methods when eigendecomposition became easier; then further replaced by normal theory MLE.

• Wilcoxon test a nonparametric version of t-test, invented to circumvent t-test computations gave rise to nonparametric statistics

• irony of nonparametric computations (sorting, etc.) implementation on a computer

• multiple regression—stepwise, all subsets tailored to available computing power

## Asymptotic inference vs computational inference

Statistical theory <=> Statistical computing

Exploration by simulation

Finite-sample explorations by Monte Carlo

Computational Statistics: computational methods that enable statistical methods (numerical analysis, graphics, database methods, software engineering, etc.)

modern statistical algorithms have to deal with complicated models, large, unstructured heterogeneous data sets (cases, variables), more graphics, more simulations

**Aspects of Computational Inference**:

Cross validation

Resampling

Simulation of data-generating processes to assign confidence level to decisions

Visualization methods

# Quadrature vs Monte Carlo Integration

- Statistical parameters are often integrals (e.g., mean)
- Parameter estimation is a problem of estimating an integral (of dimension $d$ from $n$ observations)
- Quadrature and Monte Carlo are two alternatives
- The Law of Large Numbers (LLN) states that if $X_1, X_2, \ldots, X_n$ are independent and identically distributed (i.i.d.) random variables with mean $\mu$, then $\bar{X}_n$ converges to $\mu$. Weak and Strong LLN assert different kinds of convergence.
- Monte Carlo procedure generates i.i.d. samples.
- A Monte Carlo estimate of the integral $\int f(x)g(x)dx$, (which is a parameter), the mean of $f(X)$ with respect to the density $g(x)$ is $\frac{1}{n}\sum_{i=1}^{n} f(X_i)$, where $X_1, X_2, \ldots, X_n$ are random samples from the density $g(x)$.

- The Central Limit Theorem (CLT) tells us how the distribution of $\bar{X}_n$ can be approximated for large $n$, say, if the variance of the $X_i$s is $\sigma^2$. It is done by using $\frac{\sqrt{n}(\bar{X}_n-\mu)}{\sigma}$ as $\mathcal{N}(0,1)$.

- CLT implies that the error estimate is independent of $d$

- Error: Monte Carlo: $O(n^{-1/2})$;
  Quadrature: $O(n^{-k})$, $k \geq 2$

- variance reduction in Monte Carlo, e.g., importance sampling

- Monte Carlo cost reduction: sample re-use methods; start with small $m$ and increase over iterations

- Monte Carlo should have tougher convergence

Monte Carlo optimization: explores the whole function not just find extremum

for example, if you generate from a posterior distribution by say Monte Carlo, the histogram is an estimate of the density and not just the posterior mode

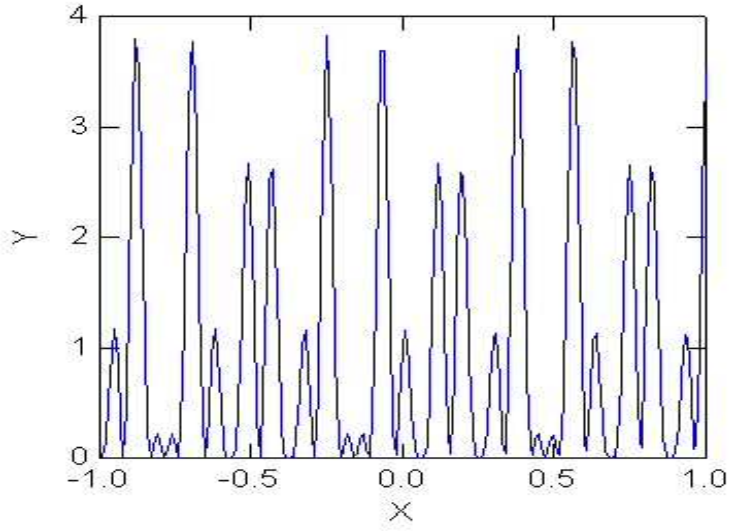
Example of Monte Carlo Optimization

by exploration of whole function

To find $\max_{\{x\in(0,1)\}} [cos(50x) + sin(20x)]^2$; Maximum is $\approx 3.705$

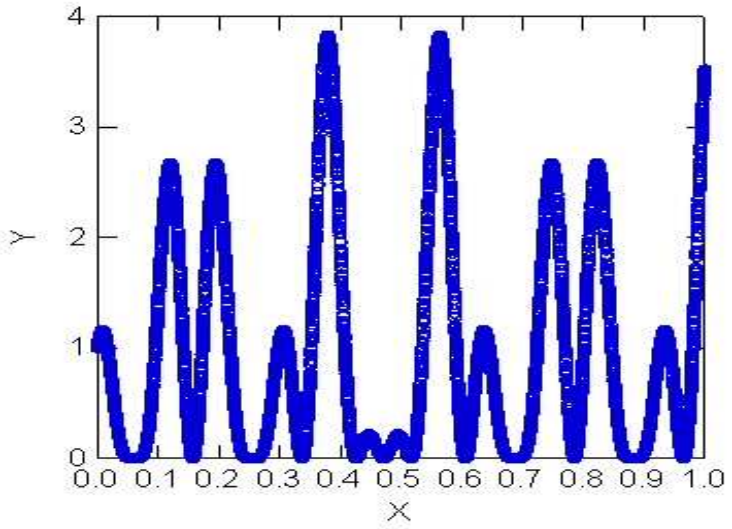
Plot of actual function

$[cos(50x) + sin(20x)]^2$



Monte Carlo approximation to $[cos(50x) + sin(20x)]^2$

based on 10000 uniform(0, 1) random samples



# Monte Carlo Integration and Optimization

- Optimization: likelihood approach
- Integration: estimation; Bayesian paradigm
- Bayesian estimation + squared error loss $\Longrightarrow$ posterior expectation $\Longrightarrow$ integration
- Bayesian estimation + general loss functions $\Longrightarrow$ minimization
- Analytically intractable $\Longrightarrow$ Monte Carlo
- Quadrature methods: curse of dimensionality
- if $n$ points in 1-dim, $n^d$ in $d$-dim
- perform best for smooth functions

## Monte Carlo Integration estimator

$x_1, x_2, \ldots, x_n$ generated from $f(x)$

$$E_f[h(X)] = \int h(x)f(x)dx$$

estimated

$$\widehat{I}_n = \frac{1}{n}\sum_{i=1}^{n} h(x_i)$$

By SLLN $\widehat{I}_n$ converges almost surely to $E_f[h(x)]$
standard error of the estimate is

$$\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}[h(x_i) - \widehat{I}_n]^2}$$

properties needed of random number generators—high periodicity, equidistribution (uniform distribution) in a high-dimensional space

Current favorite: Mersenne-Twister algorithm: pseudorandom number generator developed by Makato Matsumoto and Takuji Nishimura in 1998. Its periodicity is $2^{19937} - 1$ and has 623-dimensional equidistribution property.

## Brief Introduction to Bayesian Computation

Parameter $\boldsymbol{\theta}$. Prior: $f(\boldsymbol{\theta})$
Data: $\boldsymbol{x}$ from $f(\boldsymbol{x}|\boldsymbol{\theta})$
Posterior: $f(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{f(\boldsymbol{\theta},\boldsymbol{x})}{f(\boldsymbol{x})}$

$$= \frac{f(\boldsymbol{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{\int f(\boldsymbol{x}|\boldsymbol{\theta})f(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

Integral in denominator can be complicated
To explore posterior often MCMC (say Gibbs Sampler) is used. These Bayesian computations could be quite complex.

# References:


1. M.Altman, J.Gill & M.P.McDonald (2004): *Numerical Issues in Statistical Computing for the Social Scientist*. New York: John Wiley.

2. J.M.Chambers (1982): Algorithms, Statistical. In S.S.Kotz & N.L.Johnson (Eds.) (1982): *Encyclopedia of Statistical Sciences, Vol. 1*. New York: John Wiley. pp. 41–47.

3. J.E.Gentle (2002): *Elements of Computational Statistics*. New York: Springer-Verlag.

4. J.E.Gentle (1998): *Numerical Linear Algebra with Applications in Statistics*. New York: Springer-Verlag.

5. J.E.Gentle, W.Härdle & Y.Mori (2004): *Handbook of Computational Statistics: Concepts and Methods*. New York: Springer-Verlag.

6. W.J.Kennedy & J.E.Gentle (1990): *Statistical Computing*. (Statistics, a Series of Textbooks and Monographs). New York: Marcel-Dekker.

7. D.E.Knuth (1997): *The Art of Computer Programming: Volume I: Fundamental Algorithms*. Third edition. Reading, MA: Addison-Wesley.

8. K.Lange (1999): *Numerical Analysis for Statisticians*. New York: Springer-Verlag.

9. K.Lange (2004): *Optimization*. New York: Springer-Verlag.

10. J.F.Monahan (2001): *Numerical Methods of Statistics*. London: Cambridge University Press.

11. R.A.Thisted (1988): *Elements of Statistical Computing: Numerical Computation*. New York: Chapman & Hall.