BAYESPACK: A Collection of Numerical Integration Software for Bayesian Analysis

Alan Genz [*]
Department of Mathematics
Washington State University
Pullman, WA 99164-3113
AlanGenz@wsu.edu

Robert E. Kass
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Kass@stat.cmu.edu

October, 1989

Abstract

A software package is described for the numerical evaluation of integrals that arise in Bayesian statistical analysis. Different types of transformations that can be user selected to precondition the problem are discussed. These all begin with a standardizing transformation, that can be adaptively constructed, and is followed by a multivariate Normal, multivariate Student's t or split-t transformation. Numerical integration methods that are currently part of the package are Monte-Carlo, quasi Monte-Carlo, subregion adaptive, stochastic spherical-radial and product Gauss-Hermite and generalized Gauss-Hermite. Illustrative numerical results for examples from Bayesian statistics applications are given.

Key Words: Numerical integration, Monte Carlo, multiple integrals, Bayesian computation.

1 Introduction

The BAYESPACK software was developed for the numerical evaluation of integrals in the form

$$I(g) = \int_{R^m} g(\boldsymbol{\theta})\tilde{L}(\boldsymbol{\theta})d\boldsymbol{\theta},$$

where $\boldsymbol{\theta}$ is an m-dimensional vector of integration variables, and $\tilde{L}(\boldsymbol{\theta})$ is proportional to a posterior density function that has a dominant peak at its mode $\boldsymbol{\mu}$. It assumed that essentially all of the contribution to the integral comes from values of the integrand in some neighborhood of $\boldsymbol{\mu}$. In typical applications $g(\boldsymbol{\theta})$ is some elementary real-valued function (e.g., $g(\boldsymbol{\theta}) = \theta_j$, a component of $\boldsymbol{\theta}$) and there are often many such integrals, for various alternative choices of $g(\boldsymbol{\theta})$, that need to be evaluated, so that the main quantities of interest are expected values of the form $I(g)/I(1)$, because $\tilde{L}(\boldsymbol{\theta})$ is unnormalized. It is also assumed that the integration region $R^m = (-\infty, \infty)^m$. For problems where this is not the case, initial transformations of some or all of the integration variables will be necessary to place the problem into this standard form. Note: in typical applications the normalizing constant for the posterior is $I(1)$, assuming $\tilde{L}(\boldsymbol{\theta}) = p(\mathbf{y}/\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, where $p(\mathbf{y}/\boldsymbol{\theta})$ is the density for the data, and $\pi(\boldsymbol{\theta})$ is the prior density.

The software in the BAYESPACK has been implemented in double precision FORTRAN 77 with driver subroutines BAYSNT (short driver with defaults for some input parameters) or BANINT (long driver). If we denote the posterior mean by $\tilde{\boldsymbol{\mu}} = I(\boldsymbol{\theta})/I(1)$, and posterior covariance matrix by $\tilde{\Sigma} = I(\boldsymbol{\theta}\boldsymbol{\theta}^t)/I(1) - \tilde{\boldsymbol{\mu}}\tilde{\boldsymbol{\mu}}^t$,

[*] Supported by NATO Collaborative Research Grant CRG 940139 and US NSF and NIH grants.

the default outputs from BAYSNT and BANINT are approximate values for $I(1)$ and $\tilde{\boldsymbol{\mu}}$ (and also $\tilde{\Sigma}$ for BANINT), with estimates for errors in $I(1)$ and $\tilde{\boldsymbol{\mu}}$. A user may also request additional posterior means by defining a subroutine that computes a set of additional functions $\{g_i(\boldsymbol{\theta})\}$. In this case BAYSNT and BANINT will also return approximate values for $\{I(g_i)/I(1)\}$ with error estimates. The use BAYSNT or BANINT is described later in this paper. These descriptions are preceded with descriptions of the integration methods and the transformation methods that are implemented in the software.

2 The Integration Methods

There are three general types of integration methods implemented in BAYESPACK. The methods are classified according to the type of integration region that they were originally developed for. U^m methods are defined to be methods for the unit hypercube $U^m = [0,1]^m$, and require a transformation from R^m to U^m before they can be used. The U^m methods currently implemented are a simple Monte-Carlo method with antithetic variates, a method that uses a sequence of increasing order randomized lattice rules (see Davis and Rabinowitz, 1984) and a subregion adaptive method (a modified version of the DCUHRE subroutine developed by Berntsen, Espelid and Genz, 1991).

The second type of method, R^m methods, are defined to be methods that can be used directly for integration over R^m. The one R^m method currently implemented uses an adaptive algorithm combined with stochastic spherical-radial integration rules. This method was developed by Genz and Monahan (1996b) (see also Monahan and Genz, 1996). A third type of methods, Normal-R^m methods, are defined to be methods for numerical integration over R^m, with the multivariate normal weight function $(\sqrt{2\pi})^{-m}e^{-\frac{1}{2}\boldsymbol{\theta}^t\boldsymbol{\theta}}$. Three Normal-$R^m$ methods are implemented, a method that uses a sequence of increasing degree product Gauss-Hermite rules (see Davis and Rabinowitz, 1984), a method that uses a sequence of increasing degree rules developed by Genz and Keister (1996) and a method that uses degree three stochastic radial-spherical rules developed by Genz and Monahan (1996a).

All of the integration methods used in BAYESPACK allow input and simultaneous approximate integration of several different g functions. The methods also provide error estimates for the final integral estimates. The BAYESPACK software is designed so that all of the integration methods have a standard interface. New methods can easily be added and used with the rest of the software.

3 Transformations

3.1 The Standardizing Transformation

The integration methods in BAYESPACK use transformations of the initial integration problem(s). All of the transformations for BAYESPACK begin with a standardizing transformation of the posterior density $\boldsymbol{\theta} = \boldsymbol{\mu} + C\mathbf{y}$. Here C is the lower triangular Cholesky factor of the modal covariance matrix Σ ($\Sigma = CC^t$), the inverse of the negative of Hessian matrix for $\log(\tilde{L}(\boldsymbol{\theta}))$ at $\boldsymbol{\mu}$. The mode $\boldsymbol{\mu}$ and modal covariance matrix Σ are determined using numerical optimization for $\boldsymbol{\mu}$, and extrapolated finite difference approximations for the partial derivatives needed for Σ. After applying the standardizing transformation, we have

$$I(g) = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} g(\boldsymbol{\theta})\tilde{L}(\boldsymbol{\theta})d\boldsymbol{\theta} = |C| \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} g(\boldsymbol{\mu} + C\mathbf{y})\tilde{L}(\boldsymbol{\mu} + C\mathbf{y})d\mathbf{y}.$$

In order to avoid the underflow problems that often arise with practical posterior densities, the BAYESPACK computations use $\tilde{L}(\boldsymbol{\theta})/\tilde{L}(\mu))$ during the numerical integrations. Part of the problem input for BAYESPACK is a function to compute $\log(\tilde{L}(\boldsymbol{\theta}))$. This scaling of the problem is accomplished by adding $-\log(\tilde{L}(\mu))$ to

$\log(\tilde{L}(\boldsymbol{\theta}))$ for each $\boldsymbol{\theta}$ value that is used by the numerical integrations. Typical applications require normalized results in the form $I(g)/I(1)$ anyway, so this scaling does not theoretically affect these computations. The final unscaled value for $I(1)$ is always output.

Because of heavy and/or skewed tail behavior in some problems, an additional optional "split" scaling of the $\mathbf{y}$ integration is possible. This split scaling is implemented by an additional transformation of the form $\mathbf{y} = D(\mathbf{w})$, where $D(\mathbf{w})$ is a diagonal matrix function with entries $d_{ii}(\mathbf{w})$. These are defined by $d_{ii}(\mathbf{w}) = w_i(1 - \delta_i^- w_i)/(1 - w_i)$ if $w_i < 0$ and $d_{ii}(\mathbf{w}) = w_i(1 + \delta_i^+ w_i)/(1 + w_i)$ if $w_i \geq 0$ for $1 \leq i \leq m$. A method for determining the $2m$ parameters $\{\delta_i^{\pm}\}$ is described in Genz and Kass (1996). After this transformation, the original integral becomes

$$I(g) = |C| \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} |D'(\mathbf{w})| g(\boldsymbol{\mu} + CD(\mathbf{w}))\tilde{L}(\boldsymbol{\mu} + CD(\mathbf{w}))d\mathbf{w}.$$

R^m integration methods do not require any additional transformations and can be applied directly to these integrals. Normal-R^m integration methods assume a multivariate Normal weight. For these methods, the multivariate Normal factor is introduced at this point, and we have

$$I(g) = (\sqrt{2\pi})^{-m} \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} e^{-\frac{1}{2}\mathbf{w}^t\mathbf{w}} (\sqrt{2\pi})^m e^{\frac{1}{2}\mathbf{w}^t\mathbf{w}} |C||D'(\mathbf{w})| g(\boldsymbol{\mu} + CD(\mathbf{w}))\tilde{L}(\boldsymbol{\mu} + CD(\mathbf{w}))d\mathbf{w},$$

so that the input integrand for the Normal-R^m methods takes the form

$$f(\mathbf{w}) = (\sqrt{2\pi})^m e^{\frac{1}{2}\mathbf{w}^t\mathbf{w}} |C||D'(\mathbf{w})| g(\boldsymbol{\mu} + CD(\mathbf{w}))\tilde{L}(\boldsymbol{\mu} + CD(\mathbf{w})).$$

3.2 The Multivariate Normal Transformation

U^m integration methods of assume $[0, 1]$ integration limits for all of the variables, and therefore additional transformations are needed. The simplest option assumes that the posterior density is approximately multivariate Normal, so that after the standardizing and scaling transformations, and univariate Normal transformations of the form $w_i = \Phi^{-1}(z_i)$, $I(g)$ becomes

$$I(g) = \int_0^1 ... \int_0^1 g(\boldsymbol{\mu} + CD(\mathbf{w}(\mathbf{z}))h(\mathbf{z})d\mathbf{z},$$

where $\mathbf{w}(\mathbf{z}) = (\Phi^{-1}(z_1), ..., \Phi^{-1}(z_m))^t$ and $h(\mathbf{z}) = |C||D'(\mathbf{w}(\mathbf{z}))|(\sqrt{2\pi})^m e^{\frac{1}{2}\mathbf{w}(\mathbf{z})^t\mathbf{w}(\mathbf{z})} \tilde{L}(\boldsymbol{\mu} + CD(\mathbf{w}(\mathbf{z})))$.

3.3 The Multivariate Student's t Transformation

Another general model (see Evans and Swartz, 1995) that can be used for the posterior density is the multivariate Student's t density $K_\nu^{(m)} |\Sigma|^{-\frac{1}{2}} (1 + \frac{(\boldsymbol{\theta}-\boldsymbol{\mu})^t\Sigma^{-1}(\boldsymbol{\theta}-\boldsymbol{\mu})}{\nu})^{-\frac{\nu+m}{2}}$, where $K_\nu^{(m)} = \Gamma(\frac{\nu+m}{2})/(\Gamma(\frac{\nu}{2})(\sqrt{\nu\pi})^m)$. In this case, the modal covariance matrix (the inverse of the the negative of the Hessian matrix H for the model at $\boldsymbol{\theta} = \boldsymbol{\mu}$), and the actual covariance matrix Σ differ by a factor of $(\nu + m)/\nu$, so that the correct standardizing transformation matrix is $C = \sqrt{\frac{\nu+m}{m}}\hat{C}$, where $H = -(\hat{C}\hat{C}^t)^{-1}$. After the transformation $\boldsymbol{\theta} = \boldsymbol{\mu} + C\mathbf{y}$, and a further optional scaling transformation $\mathbf{y} = D(\mathbf{w})$, $I(g)$ becomes

$$I(g) = |C| K_\nu^{(m)} \int\limits_{-\infty}^{\infty} ... \int\limits_{-\infty}^{\infty} (1 + \frac{\mathbf{w}^t\mathbf{w}}{\nu})^{-\frac{\nu+m}{2}} (K_\nu^{(m)})^{-1} (1 + \frac{\mathbf{w}^t\mathbf{w}}{\nu})^{\frac{\nu+m}{2}} |D'(\mathbf{w})| g(\boldsymbol{\mu} + CD(\mathbf{w}))\tilde{L}(\boldsymbol{\mu} + CD(\mathbf{w}))d\mathbf{w}.$$

The density is required in a variables-separated form for the U^m methods. This form is obtained using the transformations $w_i = u_i(\prod_{j=1}^{i-1}(\nu + j - 1 + u_j^2)/(\nu + j))^{\frac{1}{2}}$ for $i = 1, 2, ..., m$. Then $I(g)$ becomes

$$K_\nu^{(1)} \int\limits_{-\infty}^{\infty} (1 + \frac{u_1^2}{\nu})^{-\frac{\nu+1}{2}} ... K_{\nu+m-1}^{(1)} \int\limits_{-\infty}^{\infty} (1 + \frac{u_m^2}{\nu + m - 1})^{-\frac{\nu+m}{2}} g(\boldsymbol{\mu} + CD(\mathbf{w}(\mathbf{u})))h(\mathbf{w}(\mathbf{u}))d\mathbf{u},$$

where $\mathbf{w}(\mathbf{u}) = (u_1, u_2((\nu+u_1^2)/(\nu+1))^{\frac{1}{2}}, ..., u_m(\prod_{j=1}^{m-1}(\nu+j-1+u_j^2)/(\nu+j))^{\frac{1}{2}})^t$ and $h(\mathbf{w}(\mathbf{u})) = |C||D'(\mathbf{w}(\mathbf{u}))|$ $(K_\nu^{(m)})^{-1}(1+\frac{\mathbf{w}(\mathbf{u})^t\mathbf{w}(\mathbf{u})}{\nu})^{\frac{\nu+m}{2}}\tilde{L}(\boldsymbol{\mu}+CD(\mathbf{w}(\mathbf{u})))$. The final transformations to the [0,1] variables z_i use the variables $u_i = t_{\nu+i-1}^{-1}(z_i)$ for $i = 1, 2, ..., m$, with $t_\nu(u) = K_\nu^{(1)} \int_{-\infty}^u (1+s^2/\nu)^{-(\nu+1)/2} ds$, and the final form for $I(g)$ is

$$I(g) = \int_0^1 ... \int_0^1 g(\boldsymbol{\mu}+CD(\mathbf{w}(\mathbf{u}(\mathbf{z}))))h(\mathbf{w}(\mathbf{u}(\mathbf{z})))d\mathbf{z},$$

where $\mathbf{u}(\mathbf{z}) = (t_\nu^{-1}(z_1), ..., t_{\nu+m-1}^{-1}(z_m))^t$.

3.4 The Split-t Transformations

For the split-t option (see Genz and Kass, 1996, for more details), the univariate Normal transformations $z_i = \Phi(w_i)$ used with the multivariate Normal transformation method are modified. These univariate transformations are split at the origin and replaced by (possibly) different transformation functions, chosen from the Student's t family, for each of the $\mathbf{w}$ variables. With this option, during initialization the function $\tilde{L}(\boldsymbol{\theta})$ is investigated along both the positive and negative directions for each of the $\mathbf{w}$ variables in order to determine the appropriate degrees of freedom $\nu_i^\pm$ and scale factors $\delta_i^\pm$, for $i = 1, 2, ..., m$ for the transformations. The split transformations take the form

$$w = T_i^{-1}(z) = \begin{cases} t_{\nu_i^\pm(z)}^{-1} & \text{if } \pm(z-0.5) > 0 \text{ and } \nu_i^\pm < \nu_{max} \\ \Phi^{-1}(z) & \text{if } \pm(z-0.5) > 0 \text{ and } \nu_i^\pm = \nu_{max} \end{cases}$$

with $w \in [0,1]$ and $\nu_{max} = 10$. The transformed integral over the unit m-cube is

$$I(g) = \int_0^1 ... \int_0^1 g(\boldsymbol{\mu}+CD(\mathbf{w}(\mathbf{z})))h(\mathbf{z})d\mathbf{z},$$

where $\mathbf{w}(\mathbf{z}) = (T_1^{-1}(z_1), ..., T_m^{-1}(z_m))^t$ and $h(\mathbf{z}) = |C||D'(\mathbf{w}(\mathbf{z}))|(\prod_{i=1}^m J_i(w_i))\tilde{L}(\boldsymbol{\mu}+CD(\mathbf{w}(\mathbf{z})))$, with the transformation Jacobian factors J_i defined by

$$J_i(w_i) = \begin{cases} (K_{\nu_i^\pm}^{(1)})^{-1}(1+w_i^2/\nu_i^\pm)^{(\nu_i^\pm+1)/2} & \text{if } \pm w_i > 0 \text{ and } \nu_i^\pm < \nu_{max} \\ \sqrt{2\pi}e^{w_i^2/2} & \text{if } \pm w_i > 0 \text{ and } \nu_i^\pm = \nu_{max}. \end{cases}$$

4 Use of BAYESPACK

4.1 Parameters for BANINT

The software described in the previous sections has been implemented in double precision FORTRAN 77 with main driver subroutine **BANINT**. A short driver **BAYSNT** with an abbreviated parameter list is also available. BAYSNT is described in the next section. The default outputs from BANINT are approximate values for $I(1)$, $\tilde{\boldsymbol{\mu}}$ and $\tilde{\Sigma}$, with estimates for errors in $I(1)$ and $\tilde{\boldsymbol{\mu}}$. The user can also request additional means by specifying additional functions $g_i(\boldsymbol{\theta})$. In this case BANINT will also return approximate values for $I(g_i)/I(1)$ with error estimates. BANINT has the calling sequence:

```
    CALL BANINT( M, RELREQ, MAXVLS, RS, USRLGP, MN, USRMNS, PROBLM, MU, C, PU,
                   NUMTRN, METHOD, NRMCON, MEANS, ERRORS, COVRNC, INFORM )
```

The parameters for BANINT are defined in the following two subsections.

4.1.1 Input Parameters

M integer number of integration variables m (the dimension of $\boldsymbol{\theta}$), with $1 \leq \mathrm{M} \leq 20$.

RELREQ real requested relative accuracy; this parameter is used as a termination parameter for the selected numerical integration method. The numerical integration proceeds until the relative errors in all of the requested $I(g)$ computations are less than RELREQ, or further work would require more than MAXVLS posterior function values.

MAXVLS integer limit on the number of posterior function values allowed.

RS integer restart parameter.

If RS = 0, then it is assumed that this is the first call of BANINT and, if NUMTRN > 0, transformation parameters are determined.

If RS > 0, then a continuation call of BANINT is assumed (typically with a larger value for MAXVLS), and transformation parameters from the previous call are used.

USRLGP name of an external real function for computation log of the posterior.

It must have one input parameter, a real M-vector that defines the evaluation point $\boldsymbol{\theta}$.

MN integer length of final MEANS vector, with MN ≥ M.

USRMNS name of an external subroutine for computation of the posterior means. It is used when MN > M and the user require additional means. USRMNS allows the user to specify additional $g_i(\boldsymbol{\theta})$ functions for computation of the means $I(g_i)/I(1)$ for $i = 1, 2, ...,$MN-M. USRMNS has two parameters: a real input M-vector (the evaluation point $\boldsymbol{\theta}$), and a real (MN-M)-vector that returns the g_i values at $\boldsymbol{\theta}$.

PROBLM character string for the name of the problem.

MU real M-vector of starting values for posterior mode optimization.

C real M(M+1)/2-vector, the lower triangular standardizing transformation array C, stored by rows, and used for input only when RS > 0 or NUMTRN = 0.

PU integer Fortran output unit number.

If PU = 0, then no output is produced.

If PU > 0, transformation information, MEANS and ERRORS are sent to unit number PU.

If PU < 0, modal covariance Cholesky factor, modal covariance matrix, transformation information, MEANS, ERRORS, and final covariance matrix are sent to unit number |PU|.

NUMTRN integer used to control types of transformations used.

If NUMTRN = 0, no transformation is used.

If 0 < NUMTRN < 9, a standardizing transformation, with (for METHOD < 20) multivariate Student's t with ν = NUMTRN.

If NUMTRN = 10, a standardizing transformation, with (for METHOD < 20) multivariate Normal.

If NUMTRN = 20, a scaled standardizing transformation, and with split-t for METHOD < 20.

METHOD integer method type:

If METHOD = 0, a Monte-Carlo importance sampling method.

If METHOD = 1, a randomized lattice rule method.

If METHOD = 10, a subregion adaptive method.

If METHOD = 20, a mixed stochastic spherical-radial method.

If METHOD = 30, a Gauss-Hermite product rules method.
If METHOD = 31, a modified Gauss-Hermite rules method.
If METHOD = 32, a stochastic radial-spherical method.

4.1.2 Output Parameters

RS integer restart parameter; RS = 1 on exit.

MU real M-vector for computed posterior mode.

C real M(M+1)/2-vector, the lower triangular standardizing transformation array, stored by rows. When RS > 0 and NUMTRN > 0, this is the Cholesky factor of the modal covariance matrix.

NRMCON real estimated value for $I(1)$, the normalizing constant, assuming $\tilde{L}(\boldsymbol{\theta}) = p(\mathbf{y}/\boldsymbol{\theta})\pi(\boldsymbol{\theta})$, where $p(\mathbf{y}/\boldsymbol{\theta})$ is the density for the data, and $\pi(\boldsymbol{\theta})$ is the prior density.

MEANS real MN-vector of estimates for posterior means.

ERRORS real MN-vector of estimated errors for MEANS.

COVRNC real lower left of approximate $\tilde{\Sigma}$ matrix stored by rows as a long vector of length M(M+1)/2.

INFORM integer completion information parameter:
If INFORM = 0, then integration was completed with estimated relative accuracy ≤ RELREQ.
If INFORM = 1, then integration was not completed with estimated relative accuracy ≤ RELREQ.

4.1.3 Example Use of BANINT

The use of BAYESPACK is illustrated in this section with the posterior density for the Stanford heart transplant data (see Genz and Kass, 1996, for details). Here is a sample program for this problem.

```
      PROGRAM HRTEST
      EXTERNAL HRLPST, HRGFNS
*
*        Stanford Heart Data Test Program
*
      CHARACTER*20 PROBLM
      PARAMETER ( PROBLM = 'Stanford Heart Data' )
      INTEGER M, MN, RS, MAXVLS, INFORM, I, PU
      PARAMETER ( M = 3, MN = 6, MAXVLS = 10000, PU = 6 )
      DOUBLE PRECISION NRMCON, MEANS(MN), ERRORS(MN), COVRNC(M*(M+1)/2)
      DOUBLE PRECISION MU(M), C( M*(M+1)/2 )
      DATA RS, MU / 0, 3.39D0,  -9.24D-2,  -7.23D-1 /
      CALL BANINT( M, 1D-2,   MAXVLS, RS, HRLPST, MN, HRGFNS,
     &                PROBLM, MU, C, PU, 20, 10,
     &                NRMCON, MEANS, ERRORS, COVRNC, INFORM )
      CALL BANINT( M, 1D-2, 2*MAXVLS, RS, HRLPST, MN, HRGFNS,
     &                PROBLM, MU, C, PU, 20, 10,
     &                NRMCON, MEANS, ERRORS, COVRNC, INFORM )
      END
*
```

```
      SUBROUTINE HRGFNS( X, GFUNS )
*
*     For computing g function values for extra posterior means.
*
      DOUBLE PRECISION X(*), GFUNS(*)
      GFUNS(1) = EXP(X(1))
      GFUNS(2) = EXP(X(2))
      GFUNS(3) = EXP(X(3))
      END
```

The original problem $[0, \infty)$ variables λ, τ and p were transformed to $(-\infty, \infty)$ variables using a log transformation for the posterior integrations, so HRGFNS allows posterior means for the original variables to be computed. The log posterior function HRLPST is given in the Appendix. The transformation type requested is split-t (NUMTRN = 20) and the method is subregion adaptive (METHOD = 10). The resulting run pruduced the following information (sent to the usual default Fortran standard output unit 5):

```
 Numerical Integration Results for Stanford Heart Data
  Laplace Approx. for Constant is  3.949-164
 Log Posterior Max    Mode Vector
     -375.304      3.38503 -0.09242 -0.72288
        Transformation Types
 1:Nrml,Nrml 2:Nrml,Nrml 3:Nrml,Nrml
        Delta Scale Factors
 1: 1.0, 1.0 2: 0.9, 1.0 3: 1.0, 1.0
    Subregion Adaptive Integration Results
                   Means
Num Pts  Constant   1        2        3
   9877 4.089-164  3.37086 -0.05164 -0.73586
Errests 7.379-167  0.00599  0.00289  0.00270
  Additional Means and Errors
   1        2        3
 32.61352  1.04385  0.49739
  0.06480  0.00909  0.00044
                   Means
Num Pts  Constant   1        2        3
  19920 4.092-164  3.36999 -0.05103 -0.73669
Errests 3.685-167  0.00209  0.00072  0.00164
  Additional Means and Errors
   1        2        3
 32.60991  1.04571  0.49713
  0.02605  0.00295  0.00016
```

The two sets of "Means" results correspond to separate calls to BANINT. For the first call, the standardizing transformation was computed after the numerical determination of the posterior mode. The split-t algorithm decided that all the integration variables were Normal for both sign possibilities and the δ parameters were all = 1.0, except for the second variable, where 0.9 was determined for the negative w_2 values. The Laplace approximation result is $|C|(\sqrt{2\pi})^m \tilde{L}(\boldsymbol{\mu})$. The "Num Pts" column refers to the number of $\tilde{L}$ values used for the numerical integration and the "Constant" column refers to the approximate value for $I(1)$. The "Errests" row is a vector of error estimates for "Means", from the numerical integration method.

The software can also accommodate the use of an “adaptive” standardizing transformation (see Evans and Swartz, 1995). With this option, both $\boldsymbol{\mu}$ and C are adjusted dynamically as the numerical integration proceeds. Defining the Cholesky factor $\tilde{C}$, for $\tilde{\Sigma}$, by $\tilde{\Sigma} = \tilde{C}\tilde{C}^t$, once approximate values for $\tilde{\boldsymbol{\mu}}$ and $\tilde{C}$ are available, these approximations are used instead of $\boldsymbol{\mu}$ and C, and dynamically updated at various stages in the numerical integration. The use of an adaptive standardizing transformation is not directly compatible with the split transformations, and therefore should not be used with the split transformations. This is illustrated with a different integration method (METHOD = 20) and by changing the transformation type to 10. If four lines of code are introduced to modify C using an approximation to $\tilde{\Sigma}$ from first call of BANINT (CHOLSK computes the Cholesky decomposition for $\tilde{\Sigma}$ found in the BANINT output parameter COVRNC), then the original calls to BANINT now become

```
      CALL BANINT( M, 1D-2,   MAXVLS, RS, HRLPST, MN, HRGFNS,
     &                PROBLM, MU, C, PU, 10, 20,
     &                NRMCON, MEANS, ERRORS, COVRNC, INFORM )
      DO I = 1, M*(M+1)/2
         C(I) = COVRNC(I)
      END DO
      CALL CHOLSK( M, C )
      CALL BANINT( M, 1D-2, 2*MAXVLS, RS, HRLPST, MN, HRGFNS,
     &                PROBLM, MU, C, PU, 10, 20,
     &                NRMCON, MEANS, ERRORS, COVRNC, INFORM )
```

The output results are:

```
 Numerical Integration Results for Stanford Heart Data
  Laplace Approx. for Constant is  3.949-164
 Log Posterior Max    Mode Vector
     -375.304      3.38503 -0.09242 -0.72288
      Stochastic Radial Spherical Results
                   Means
Num Pts  Constant   1        2        3
   9760 4.099-164  3.37161 -0.05059 -0.73774
Errests 5.721-167  0.00536  0.00066  0.00132
  Additional Means and Errors
   1        2        3
 32.66887  1.04553  0.49781
  0.08209  0.00289  0.00113
                   Means
Num Pts  Constant   1        2        3
  19660 4.097-164  3.36814 -0.05045 -0.73706
Errests 4.375-167  0.00380  0.00110  0.00130
  Additional Means and Errors
   1        2        3
 32.59738  1.04788  0.49715
  0.05315  0.00219  0.00062
```

This section is concluded with an illustration of the use of the multivariate Student’s t transformation. Changing NUMTRN to 5, the calls to BANINT are:

```
      CALL BANINT( M, 1D-2,   MAXVLS, RS, HRLPST, MN, HRGFNS,
     &                PROBLM, MU, C, PU, 6, 10,
     &                NRMCON, MEANS, ERRORS, COVRNC, INFORM )
      CALL BANINT( M, 1D-2, 2*MAXVLS, RS, HRLPST, MN, HRGFNS,
     &                PROBLM, MU, C, PU, 6, 10,
     &                NRMCON, MEANS, ERRORS, COVRNC, INFORM )
```

The output results are:

```
 Numerical Integration Results for Stanford Heart Data
  Laplace Approx. for Constant is  3.949-164
 Log Posterior Max    Mode Vector
     -375.304      3.38503 -0.09242 -0.72288
        Transformation Types
 1: T-5, T-5 2: T-6, T-6 3: T-7, T-7
    Subregion Adaptive Integration Results
                   Means
Num Pts  Constant   1        2        3
   9877 4.094-164  3.36871 -0.05073 -0.73738
Errests 7.194-167  0.00539  0.00160  0.00102
  Additional Means and Errors
   1        2        3
 32.60766  1.04703  0.49700
  0.05030  0.00119  0.00108
                   Means
Num Pts  Constant   1        2        3
  19920 4.096-164  3.36809 -0.05061 -0.73767
Errests 2.230-167  0.00119  0.00014  0.00071
  Additional Means and Errors
   1        2        3
 32.59556  1.04688  0.49690
  0.00531  0.00044  0.00018
```

4.2 BAYSNT: A Short Driver Subroutine for BANINT

A driver subroutine BAYSNT is also provided as part of BAYESPACK. This subroutine uses a subset of the parameters for BANINT and automatically provides defaults for other parameters. The parameters for BAYSNT are M, MAXVLS, USRLGP, MN, USRMNS, PU, PROBLM and MU. Default values for other input parameters are RS = 0 and NUMTRN = 20. BAYSNT calls BANINT with METHOD = 0, 10, 20, and outputs results to Fortran unit |PU|. The relevant parameter definitions from the previous section are repeated here.

M integer number of integration variables m (the dimension of $\boldsymbol{\theta}$), with $1 \leq \mathrm{M} \leq 20$.

MAXVLS integer limit on the number of posterior function values allowed.

USRLGP name of an external real function for computation log of the posterior.
It must have one input parameter, a real M-vector that defines the evaluation point $\boldsymbol{\theta}$.

MN integer length of final MEANS vector, with $\mathrm{MN} \geq \mathrm{M}$.

USRMNS name of an external subroutine for computation of the posterior means. It is used when MN > M and the user require additional means. USRMNS allows the user to specify additional $g_i(\boldsymbol{\theta})$ functions for computation of the means $I(g_i)/I(1)$ for $i = 1, 2, ...,$MN-M. USRMNS has two parameters: a real input M-vector (the evaluation point $\boldsymbol{\theta}$), and a real (MN-M)-vector that returns the g_i values at $\boldsymbol{\theta}$.

PROBLM character string for the name of the problem.

MU real M-vector of starting values for posterior mode optimization.

PU integer Fortran output unit number.
If PU = 0, then no output is produced.
If PU > 0, transformation information, MEANS and ERRORS are sent to unit number PU.
If PU < 0, modal covariance Cholesky factor, modal covariance matrix, transformation information, MEANS, ERRORS, and final covariance matrix are sent to unit number |PU|.

Here is an example program for BAYSNT for the Stanford heart transplant data posterior. HRLPST and HRGFNS are the same as in the previous examples.

```
      PROGRAM HRTEST
      EXTERNAL HRLPST, HRGFNS
*
*        Stanford Heart Data Test Program
*
      CHARACTER*20 PROBLM
      PARAMETER ( PROBLM = 'Stanford Heart Data' )
      DOUBLE PRECISION MU(3)
      DATA MU / 3.39D0,  -9.24D-2,  -7.23D-1 /
      CALL BAYSNT( 3, 10000, HRLPST, 6, HRGFNS, PROBLM, MU, 5 )
      END
```

The output results are:

```
 Numerical Integration Results for Test Problem
  Laplace Approx. for Constant is  3.949-164
 Log Posterior Max    Mode Vector
     -375.304      3.38503 -0.09242 -0.72288
        Transformation Types
 1:Nrml,Nrml 2:Nrml,Nrml 3:Nrml,Nrml
        Delta Scale Factors
 1: 1.0, 1.0 2: 0.9, 1.0 3: 1.0, 1.0
        Monte Carlo Integration Results
                   Means
Num Pts  Constant   1        2        3
  10000 4.091-164  3.36863 -0.04768 -0.73855
Errests 1.004-166  0.00781  0.00380  0.00272
  Additional Means and Errors
   1        2        3
 32.44704  1.05307  0.49583
  0.13520  0.01179  0.00148
 Numerical Integration Results for Test Problem
  Laplace Approx. for Constant is  3.949-164
 Log Posterior Max    Mode Vector
     -375.304      3.38503 -0.09242 -0.72288
        Transformation Types
 1:Nrml,Nrml 2:Nrml,Nrml 3:Nrml,Nrml
        Delta Scale Factors
 1: 1.0, 1.0 2: 0.9, 1.0 3: 1.0, 1.0
    Subregion Adaptive Integration Results
                   Means
Num Pts  Constant   1        2        3
   9877 4.089-164  3.37086 -0.05164 -0.73586
Errests 7.378-167  0.00599  0.00289  0.00270
  Additional Means and Errors
   1        2        3
 32.61352  1.04385  0.49739
  0.06480  0.00909  0.00044

 Numerical Integration Results for Test Problem
  Laplace Approx. for Constant is  3.949-164
 Log Posterior Max    Mode Vector
     -375.304      3.38503 -0.09242 -0.72288
        Delta Scale Factors
 1: 1.0, 1.0 2: 0.9, 1.0 3: 1.0, 1.0
      Stochastic Radial Spherical Results
                   Means
Num Pts  Constant   1        2        3
   9840 4.098-164  3.36703 -0.05005 -0.73828
Errests 5.533-167  0.00462  0.00057  0.00116
  Additional Means and Errors
   1        2        3
 32.60254  1.04761  0.49675
  0.06948  0.00232  0.00091
```

5 Concluding Remarks

The BANINT software has been tested using a variety of statistics integration problems (see the POSTPACK software available on the web page with url "http://www.sci.wsu.edu/math/faculty/genz/homepage"). For most of these problems exact answers are not known, but the results obtained using different methods were usually found to be consistent with the bounds provided by the error estimates for the methods. It is hoped that the BANINT software will provide efficient solutions to statistics integration problems where the dimensionality is not too high (possibly for $2 \leq m \leq 15$). The BANINT software can also provide a means of comparison with some of the currently popular Markov chain Monte-Carlo methods, where practical error estimation is sometimes difficult.

Two problems sometimes found initially when using the package are i) failure with the numerical optimization to determine the approximate $\boldsymbol{\mu}$, because of a poor starting value, and ii) numerical overflows or underflows that occur in the computation of $\log(\tilde{L})$. The first problem can usually be solved by the use of a better starting value for $\boldsymbol{\mu}$ that can come from detailed knowledge of the application problem or the use of some auxiliary numerical optimization method. The second problem can usually be solved by a careful examination of the Fortran function that is used for the definition of the $\log(\tilde{L})$, followed by minor modifications to the source code to trap for underflows and overflows.

Acknowledgment and Final Note

This work was partially supported by grants from the National Science Foundation and the National Institutes of Health. All software and documentation described in this paper and all of the following papers with Genz as a coauthor are available at the web page with url:
"http://www.sci.wsu.edu/math/faculty/genz/homepage".

References

Berntsen, J., Espelid, T.O. and Genz, A. (1991), Algorithm 698: DCUHRE– An adaptive multiple integration routine for a vector of integrals, *ACM TOMS* 17, 452–456.

Davis, P. J. and Rabinowitz P. (1984), *Methods of Numerical Integration*, Academic Press, New York.

Evans, M. and Swartz, T. (1995), Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statist. Science*, 10, 254–272.

Genz, A. and Kass, R. (1996), Subregion adaptive integration of functions having a dominant peak, to appear in *J. Comp. Graph. Stat.*.

Genz, A. and Keister, B. (1996), Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight, *J. Comp. Appl. Math.*, **71**, 299–309.

Genz, A. and Monahan, J. (1996a), Stochastic Integration Rules for Infinite Regions, to appear in *SIAM J. Sci. Com.*, **19**.

Genz, A. and Monahan, J. (1996b), A mixed adaptive algorithm for multiple integrals over infinite regions, working paper,

Monahan, J. and Genz, A. (1996), Spherical-radial integration rules for Bayesian computation, *JASA*, to appear.

Appendix: HRLPST

```
      DOUBLE PRECISION FUNCTION HRLPST(X)
*
*     To compute log posterior for Stanford Heart Transplant Data
*      Reference:
*       Naylor, J. C. and Smith, A. F. M. (1988),  "Econometric
*       Illustrations of Novel Numerical Integration Strategies for
*       Bayesian Inference", J. Econometrics, 38, pp. 103-125.
*
      DOUBLE PRECISION X(3), LAMBDA, TAU, P
      DOUBLE PRECISION LGL, LGT, LGP, FVAL, ESUM, ETSUM, EMX
      INTEGER I, NCASES
      PARAMETER ( NCASES = 82, EMX = 100 )
      INTEGER WAIT(NCASES),SURVIV(NCASES),EXACT(NCASES),TREATD(NCASES)
      SAVE TREATD, WAIT, EXACT, SURVIV
      DATA TREATD/ 30*0, 52*1/
```

```
 DATA WAIT/
&   49,    5,   17,    2,   39,   84,    7,    0,
&   35,   36, 1400,    5,   34,   15,   11,    2,
&    1,   39,    8,  101,    2,  148,    1,   68,
&   31,    1,   20,  118,   91,  427,    0,   35,
&   50,   11,   25,   16,   36,   27,   19,   17,
&    7,   11,    2,   82,   24,   70,   15,   16,
&   50,   22,   45,   18,    4,    1,   40,   57,
&    0,    1,   20,   35,   82,   31,   40,    9,
&   66,   20,   77,    2,   26,   32,   13,   56,
&    2,    9,    4,   30,    3,   26,    4,   45,
&   25,    5/
 DATA EXACT/ 10*1, 0, 16*1, 3*0, 14*1, 0, 3*1, 2*0,
&     4*1, 2*0, 1, 0, 1, 2*0, 3*1, 0, 1, 0, 1, 2*0,
&     3*1, 0, 1, 2*0, 2*1, 3*0/
 DATA SURVIV/ 30*0,
&   15,    3,
&  624,   46,  127,   61, 1350,  312,   24,   10,
& 1024,   39,  730,  136, 1379,    1,  836,   60,
& 1140, 1153,   54,   47,    0,   43,  971,  868,
&   44,  780,   51,  710,  663,  253,  147,   51,
&  479,  322,  442,   65,  419,  362,   64,  228,
&   65,  264,   25,  193,  196,   63,   12,  103,
&   60,   43/
 LGL = X(1)
 LAMBDA = EXP( MIN( MAX( LGL, -EMX ), EMX ) )
 LGT = X(2)
 TAU =    EXP( MIN( MAX( LGT, -EMX ), EMX ) )
 LGP = X(3)
 P =      EXP( MIN( MAX( LGP, -EMX ), EMX ) )
 FVAL = 0
 ESUM = 0
 ETSUM = 0
 DO I = 1, NCASES
    ESUM = ESUM + EXACT(I)
    ETSUM = ETSUM + EXACT(I)*TREATD(I)
    FVAL = FVAL + P*LOG( LAMBDA/( LAMBDA+WAIT(I)+TAU*SURVIV(I) ) )
    FVAL = FVAL - EXACT(I)*LOG( LAMBDA+WAIT(I)+TAU*SURVIV(I) )
 END DO
 HRLPST = X(1)+X(2)+X(3) + FVAL + ESUM*LGP + ETSUM*LGT
 END
```