COMPUTATION RESEARCH GROUP
Stanford Linear Accelerator Center
Stanford, California

CGTM NO. 188
June 1977

DIVONNE2

A Program for Multiple Integration and Adaptive Importance Sampling

Jerome H. Friedman
Computation Research Group

DIVONNE2 adaptively partitions a multidimensional coordinate space into a set of axis oriented hyper-rectangular regions based on a user provided function of the coordinates. The goal is to adjust the size and shape of the regions so that the range of function values within each is small. These regions are then used as the basis for a stratified sampling estimate of the integral of the function, or to sample random vectors from the coordinate space with probability density approximating the function.

Structure

Subroutine subprogram

User Entry Names: PARTN, INTGRL, USRINT, GENPNT, TREMDMP, BUKDMP, USRTRM, FUN

Files Referenced: OUTPUT, TAPE NFILE: optional user defined external file (see below)

External Routines Referenced: ARAN9(X,N)

Common Blocks: /DATE/, /PRINT/, /ISTRGE/, /RSTRGE/, /LIMITS/, /INSMPL/, /QUADRE/, /START/, /EXFILE/, /DISPOS/, /DEPTHS/, /MXMZER/, /SAMPLE/, /SPCNTL/, /MLIMIT/, /TRESZE/, /EXMCTL/, /BUKSZE/, /GENINL/, /FUNN/, /ANSWER/, /MAXERR/, /SIGSPL/, /STRAND/, /WITCH/

Usage

The function (integrand) is defined by a user coded function subprogram:

```
FUNCTION FUN(D,X)
INTEGER D
REAL X(D)
   .
   .
   .
FUN=
   .
   .
   .
RETURN
END
```

Both arguments are input to the subprogram. The first, D, is the number of integration variables (dimensionality of integration space) and the second is a vector containing the coordinates of a point in the integration volume. The function subprogram must return (in FUN) the value of the integrand for the coordinate values stored in X.

The partitioning is invoked by a subroutine call

```
CALL PARTN (ND, SPRDMX, MAXPTS)
```

All arguments are input to the subroutine and must be specified by the user.

ND = number of integration variables (dimensionality of integration space)

SPRDMX and MAXPTS control termination of the partitioning.

If SPRDMX <0, then partitioning terminates when the root sum squared spread of function values (ΔF) summed over the current set of regions is less than $|\text{SPRDMX}|$.

If SPRDMX ≥ 0, then partitioning terminates when $\Delta F/|I| \leq$ SPRDMX. Here I is an approximate estimate of the integral of the function maintained while partitioning.

Under usual conditions (functions that are not excessively difficult), values of SPRDMX from 1.0 to 2.0 are reasonable. for difficult functions, these values many be too optimistic.

MAXPTS = maximum number of integrand evaluations. Control is returned to the calling program at the end of the first iteration for which the number of integrand evaluations (calls to FUN) exceed MAXPTS.

To obtain a more accurate estimate of the definite integral, the subroutine call

```
CALL INTGRL(IDEG,FUNINT,ERROR)
```

must be executed after control returns from PARTN. The argument IDEG is input to INTGRL, while FUNINT and ERROR are the estimated values of the integral and associated uncertainty, respectively, as output. IDEG controls the several options supplied for estimating the integral within each hyper-rectangular region. The total integral estimate is taken as an uncorrelated sum over all regions, of the individual region estimates.

IDEG=0: quasi uniform Monte Carlo estimate

IDEG=1: pseudo random Monte Carlo estimate with importance sampling

IDEG=2: second degree quadrature formula estimate

IDEG=3: third degree quadrature formula estimate

IDEG=5: fifth degree quadrature formula estimate

IDEG<0: estimate by user provided procedure (see below)

Experience indicates that the quasi uniform Monte Carlo (IDEG=0) tends to give the most accurate integral estimates. Uncertainty estimates (ERROR) are returned only for the Monte Carlo estimates (IDEG=0,1).

It should be noted that the function used to partition the coordinate space need not be the same as that being integrated with INTGRL. In particular, a single partitioning may be used in the integration of several not too dissimilar functions by repeated calls to INTGRL. If the integrand is to change between such calls, there must be user coded logic in both the calling program and FUN, to ensure that the correct integrand is evaluated. Communication between the calling program and FUN can be accomplished with labeled commons.

In order to randomly draw vectors from the coordinate space with frequency approximating the user defined function, the subroutine call

```
CALL GENPNT(X,WT)
```

must be repeatedly executed after control returns from PARTN. Each such execution will store the ND coordinates of a new random point in the vector X, and a corresponding weight in WT. Applying this weight with the point X, results in distributions that are statistically equivalent to a sample of points distributed with the density coded in FUN. The user density (as coded in FUN) need not be normalized.

The above procedure assumes that the region of integration (or generation) is enclosed by the ND dimensional unit hypercube. If this is not the case, the user can declare the labeled common

```
COMMON /LIMITS/ XMINUS(MAXDIMEN), XPLUS(MAXDIMEN)
```

and store the lower and upper limit of each integration variable in the corresponding location of XPLUS and XMINUS. MAXDIMEN is the maximum allowed dimensionality defined when the DIVONNE package is compiled (currently MAXDIMEN=10).

Method

The partitioning is accomplished by a nested refinement procedure. The procedure is iterative and, at any stage, it is only concerned with one particular hyper-rectangular region. Associated with each region is a measure of the spread of function values within the region. At any stage of the partitioning, the region for which this measure is largest is the one chosen for further partitioning (initially, the whole integration region is the one under consideration).

The quantity used to characterize the variation V_i of the function values within each region, R_i, is the range or difference of extreme values

$$V_i = \max_{\vec{x}\in R_i} f(\vec{x}) - \min_{\vec{x}\in R_i} f(\vec{x}) \tag{1}$$

The extreme values are found by standard optimization algorithms. This choice of a measure transforms the estimation problem to an optimization problem. The art of optimizing a function of several variables is well developed and can be done quite reliably with relatively few function evaluations. Except for the very simplist cases, the converse is true for estimation. The quantity

$$S_i = \tfrac{1}{2} V_i \cdot Vol\ (R_i) \tag{2}$$

(called the spread) bounds the uncertainty of any estimate of the integral within the region and is used as a relative measure of the contribution of the region to the uncertainty of the global integral estimate.

The purpose of refining the region under consideration into two or more subregions is to reduce its contribution to the global uncertainty, thus reducing the global uncertainty. After the refinement, the daughter subregions are merged into the list of all regions and the one with the largest spread measure, S_i, is the next one to be considered for further refinement. This successive refinement continues until the root sum of squares of the spread values

$$S = \left[\sum_{i=1}^{n} S_i^2\right]^{\frac{1}{2}}$$

is less than the user specified limit, or until the number of function evaluations exceeds the user specified maximum.

After partitioning, the variation of the function values within each hyperrectangular region is considerably smaller than the variation of the function over the total integration volume. One can then apply standard numerical techniques to evaluate the integral within each subregion. These techniques tend to work very well when the variation of integrand values is small. The standard techniques provided in DIVONNE2 are quasi uniform and pseudo random Monte Carlo methods, and three quadrature formulas. The user can also provide additional methods at his/her discretion. Of the methods provided with DIVONNE2, the quasi uniform Monte Carlo seems to perform the best when there are not large discontinuities in the function within the region of integration. When there are, the pseudo random Monte Carlo seems to work best.

Random vectors are sampled by choosing randomly a hyper-rectangular region with probability proportional to the estimated integral within the region. A very approximate (but adequate for this purpose) estimate of the integral is maintained during partitioning. If a more accurate estimate is desired, INTGRL may be called before sampling begins. Once a random region is chosen, a vector is drawn randomly with uniform probability from within the region.

Additional Options (Bells and Whistles)

Although the minimum necessary to use the program has been described above, there are several user options that enhance the flexibility and usefulness of the program. These options are invoked by changing the value of an internal parameter stored in a labeled common. Default values for these parameters are set via a BLOCK DATA subprogram internal to the DIVONNE2 package at compilation time. They can be changed by declaring the appropriate labeled common in a user routine and reset via an executable statement.

Numerical Options

COMMON /SAMPLE/ NPOINT

As part of the partitioning procedure for each subregion, the program draws NPOINT quasi uniform vectors within each region. The vector for which the function value is largest/smallest is used as the starting point for the maximization/minimization. The average function value for these vectors is used to obtain an approximate estimate of the integral within the region. Decreasing NPOINT may reduce the number of function evaluations used in the partitioning. Increasing its value increases the robustness of the optimization procedure and increases the accuracy of the approximate integral estimate. NPOINT must be greater than two and less than a maximum value set at compile time (currently 200). The default value is 47.

COMMON /QUADRE/ IDEG

In addition to the quasi uniform estimate of the approximate integral within each region during the partitioning, the user may invoke quadrature formula estimates. These estimates may be useful for defining alternate criteria for termination of partitioning. Usually, these quadrature estimates are not useful during partitioning and simply waste function evaluations.

IDEG = 0 : no numerical quadrature

IDEG = 2 : 2nd degree quadrature within each subregion

IDEG = 3 : 2nd and 3rd degree quadrature within each subregion

IDEG = 5 : 2nd, 3rd and 5th degree quadrature within each region

Default = 0

COMMON /INSMPL/ INTPNT

INTPNT controls the number of sample points used for the final integral estimate in INTGRL when Monte Carlo sampling is used. For quasi uniform Monte Carlo (IDEG=0), INTPNT is the number of samples drawn in each region. For pseudo random Monte Carlo (IDEG=1), INTPNT times the number of regions is the total number of samples drawn. The number used in each region is proportional to the spread, S_i, within the region. For quadrature formula estimation, the number of samples drawn in each region depends upon the particular formula. For those provided here, the numbers are as follows:

2nd degree formula (IDEG=2)	ND+1 points/region
3rd degree formula (IDEG=3)	2*ND points/region
5th degree formula (IDEG=5)	2*ND**2+1 points/region

Control Options

COMMON /PRINT/ IPRINT

Controls printed output. Iteration information is printed when the iteration number is an exact multiple of IPRINT and upon termination.

default = 1 (maximum printing)

COMMON /EXFILE/ NFILE

Defines FORTRAN file number (TAPE NFILE) for external storage to save intermediate results between runs (see below).

default = 1

COMMON /DISPOS/ IDISP

Controls the storage on an external file of information necessary to continue partitioning with a future job (or job step) at the last iteration of the present one.

IDISP = 0 : no action

IDISP $\neq$ 0 just before PARTN returns control to calling program, TAPE NFILE is rewound and information is written onto TAPE NFILE for continuing the partitioning at that point at which it left off.

default = 0.

COMMON /START/ ISTART

Defines initialization state at time of call to PARTN.

ISTART = 1 : new integration, begin partitioning

ISTART = 2 : subsequent call, continue partitioning from last iteration of previous call to PARTN in this job step. This value (ISTART=2) is not valid for the first call to PARTN within a job step

ISTART = 3 : continue partitioning from last iteration of previous job (or job step) by reading information from TAPE NFILE

default = 1

Output

During the execution of the program, the user can have access to intermediate results by declaring the appropriate labeled commons. These values are the current estimates of the various quantities and are updated with each iteration. For the correct functioning of the algorithm, these quantities should never be reset by the user.

COMMON /FUNN/ NFUN

NFUN = number of integrand evaluations

```
COMMON /ANSWER/ NRGN,FINTGL,SPRD,SPRDMX,MAXRGN,QUAD2,QUAD3,QUAD5
```

NRGN = number of subregions

FINTGL = approximate quasi Monte Carlo estimate of integral

SPRD = root sum squared spread

SPRDMX = largest region spread

MAXRGN = subregion number with largest spread

QUAD2 = 2nd degree quadrature estimate of integral

QUAD3 = 3rd degree quadrature estimate of integral

QUAD5 = 5th degree quadrature estimate of integral

The quadrature estimates exist only if IDEG has been set appropriately before the call to PARTN.

User Defined Termination

When invoked, PARTN returns control to the calling program only if the specified root sum squared spread has been reached or the specified number of integrand evaluations have been exceeded. Based on his/her own logic and the values of the various output quantities defined above (for example, consistency among the various quadrature formulae and the quasi Monte Carlo estimate), the user may wish to define other stopping criteria. This can be done by coding a user defined function subroutine.

```
LOGICAL FUNCTION USRTRM (ITER)
  .
  .
  .
RETURN
END
```

The argument is input to the routine and contains, as a value, the iteration number. Returning a value .FALSE. for USRTRM will cause continued iteration subject to the usual convergence criteria. Returning a value of .TRUE. will cause immediate termination of the partitioning procedure.

User Provided Integration Procedure

The numerical integration procedures provided with DIVONNE2, for obtaining the integral estimates within each subregion, are by no means exhaustive, and the user may wish to try others. This is accomplished by setting IDEG<0 in the call to INTGRL and by supplying a subroutine with the following header.

```
SUBROUTINE USRINT (RGNL,RGNU,FINTIN,SPRDIN,FNTOUT,ERRSQ)
```

The first four quantities are input to the subroutine and the last two are output.

RGNL(ND) = lower limits on each coordinate that define the region

RGNU(ND) = upper limits on each coordinate that define the region

FINTIN = approximate quasi Monte Carlo estimate of integral within the region obtained during partitioning

SPRDIN = function spread measure within region

FNTOUT = estimate of integral within region

ERRSQ = squared error estimate of integral evaluation within region

The total integral is taken to be the sum of the individual region estimates and the uncertainty is taken to be the square root of the sum of the squared error estimates.

ACKNOWLEDGMENT

The author wishes to thank David Carey and Fred James for helpful suggestions

APPENDIX

This section is added for completeness. It discusses internal aspects of the program that should seldom, if ever, concern the user.

Storage Management

The memory necessary to manage the partitioning and the associated hyper-rectangular regions is declared in two labeled commons:

```
COMMON /ISTRGE/ MXRGNS, ISTOR (4*MXRGNS)
COMMON /RSTRGE/ RSTSZE, RSTOR (RSTSZE)
```

The first is for storage of integer quantities and the second is for real variables. MXRGNS is the maximum number of regions for which there is sufficient storage. As distributed, the program permits a maximum of 3000 regions. As indicated, ISTOR must be dimensioned four times MXRGNS. RSTSZE is simply an integer that stores the dimension of RSTOR. If RSTSZE is not large enough for the maximum number of regions specified (MXRGNS), then the program will automatically reduce MXRGNS so that there is sufficient storage. As distributed, RSTSZE=18001, which is enough storage for 3000 regions if no quadrature formula estimation is used during partitioning.

If, during partitioning, the total number of regions is about to exceed the maximum number for which there is sufficient storage, the partitioning terminates and control is returned to the calling program. If IDISP has been set nonzero, then information for continuing the partitioning has been written to TAPE NFILE. The user may then increase the available storage by declaring the above commons in the calling program and increasing the values of MXRGNS and RSTSZE (as well as the associated dimensions of ISTOR and RSTOR), and continue the partitioning where it left off at the previous job.

Control of Recursion

The discussion of the partitioning procedure above is slightly oversimplified in that it was described as a purely iterative procedure. For efficiency considerations, the iteration is augmented by recursion. That is, after a region has been partitioned into daughter subregions, each daughter is immediately under consideration for further splitting without its being incorporated into the list of all regions and updating the list. Each daughter that is further partitioned is also under immediate consideration for further refinement and so forth. This recursion stops when all daughters have values of the spread measure less than the second largest value in the list (remember that the region with the largest value was the one chosen for partitioning) or until a maximum recursion depth is reached.

The maximum recursion depth is specified in a labeled common

```
COMMON /DEPTHS/ ISTDPH, INCDPH
```

The first integer (ISTDPH) is the maximum recursion depth for the very first region under consideration, namely, the whole integration volume. INCDPH is the maximum recursion depths for all further subregions under consideration. Default values set in the program are ISTDPH=3 and INCDPH=5.

Control of Optimization

The procedure for performing the maximizations and minimizations is described in: *S. Derenzo, "MINF-68, A general minimizing routine," Lawrence Berkeley Lab, Berkeley, Calif., Group A Tech. Rep. P-190, 1969.* There are several quantities associated with the optimization routine that can be adjusted. These appear in the following labeled common:

```
COMMON /MXMZER/ IPRNT, FSTEP, FSTRM, FFAC, NSTEP
```

IPRNT is a print flag for the optimization program.

$\text{IPRNT} \leq 0$: no optimizer printout

$\text{IPRNT} > 0$: printout every IPRNT-th optimizer iteration

default = 0

FSTEP controls the initial step size for each invocation of the optimization algorithm. The step size for each coordinate is

$$\Delta_i = FSTEP*[X_i(u) - X_i(\ell)]$$

where $X_i(u)$ and $X_i(\ell)$ are the upper and lower limit on the ith coordinate defining the subregion. The default value for FSTEP is 0.01. FSTRM and FFAC define convergence of the optimization. Convergence is achieved if for an iteration of the optimizer no value of X_i has changed more than FSTRM* Δ_i from its value at the previous iteration. The default value for FSTRM is 10^{-10}. Convergence is also achieved if the value of the function has not changed significantly from the previous iteration. Let ΔF be the absolute change of optimum function value from the previous to the present iteration. If FFAC is positive, then $\Delta F < \text{FFAC}$ defines convergence, while if FFAC is negative, then $\Delta F/|F| \leq |\text{FFAC}|$ defines convergence where F is the optimum function value at the present iteration. The default value for FFAC is -0.02.

NSTEP is the maximum number of iterations allowed on any single optimization. If the optimization has not converged after NSTEP iterations, it will stop iterating and return the best solution so far obtained. In this case, the message "EXTREEMUM NOT FOUND" will be printed (irrespective of the value of IPRNT). The default value for NSTEP is 10.

Control of Splitting Algorithm

The basic algorithm for choosing how to best partition a particular region under consideration (not described in this document) has three parameters.

These are stored in the following labeled common:

```
        COMMON /SPCNTL/ EDGFAC, CONFAC, FACCON
```

EDGFAC defines the range allowed for the location, $X_i(p)$, of a partition along each coordinate.

Let Δ_i = EDGFAC*$[X_i(u) - X_i(\ell)]$. Then the permitted range for location of a partition is $X_i(\ell) + \Delta_i \leq X_i(p) \leq X_i(u) - \Delta_i$. The default value for EDGFAC is 0.05. The last two quantities (CONFAC, FACCON) should be adjusted only after achieving a detailed understanding of the splitting algorithm. Their default values are 0.1 and 1.5 respectively.

Storage Dumps

Two subroutines are available for dumping the contents of the ISTOR and RSTOR arrays in readable formats. These may be invoked after control returns from PARTN to obtain a detailed understanding of the resulting partitioning of the integration volume.

```
                        CALL TREDMP
```

causes the binary tree associated with the paritioning to be printed, one line for each nonterminal node, with the following format

```
        NONTERMINAL NODE(I) = IM IL IR IX PRTN
```

IM = maximum region number of left subtree of node I

IL = left son of node I

IR = right son of node I

IX = coordinate upon which split was made at the Ith partitioning

PRTN = value of split point on IX-th coordinate

```
                        CALL BUKDMP
```

causes the information concerning each hyper-rectangular region to be printed. This information includes the limits on each coordinate, $X_i(\ell)$ and $X_i(u)$, that define the region, along with the approximate quasi uniform Monte Carlo estimate of the integral within the region, the spread measure squared for the region, the coordinate and split point for the next partition of the region, and the various quadrature formula integral estimates (if present).