INFORMS Journal on Computing
Vol. 24, No. 1, Winter 2012, pp. 1–9
ISSN 1091-9856 (print) | ISSN 1526-5528 (online)
http://dx.doi.org/10.1287/ijoc.1100.0416

Automating Bivariate Transformations

Jeff X. Yang, John H. Drew, Lawrence M. Leemis
Department of Mathematics, College of William & Mary, Williamsburg, Virginia 23187
{jxyang@gmail.com, jhdrew@math.wm.edu, leemis@math.wm.edu}

We automate the bivariate change-of-variables technique for bivariate continuous random variables with arbitrary distributions. This extends the algorithm for univariate change-of-variables devised by Glen et al. [Glen, A. G., L. M. Leemis, J. H. Drew. 1997. A generalized univariate change-of-variable transformation technique. *INFORMS J. Comput.* **9**(3) 288–295]. Our transformation procedure handles one–to–one, k–to–one, and piecewise k–to–one transformations for both independent and dependent random variables. We also present other procedures that operate on bivariate random variables (e.g., calculating correlation and marginal distributions).

Key words: computational probability; computer algebra systems; continuous random variables; transformation of random variables
History: Accepted by Winfried Grassmann, Area Editor for Computational Probability and Analysis; received March 2009; revised November 2009, July 2010; accepted July 2010. Published online in *Articles in Advance* December 2, 2010.

1. Introduction

Probability theory addresses a variety of problems that are both tedious and impractical to work by hand. The advent of computer algebra systems, such as Maple and Mathematica, has facilitated the development of probability packages. These packages are implementations of algorithms that can quickly perform the required tedious calculations. One such package, A Probability Programming Language (APPL), coded in Maple, performs standard probability operations on univariate random variables (Glen et al. 2001). Many problems that would have taken hours to solve by hand can be quickly computed in APPL via a few lines of code.

Consider the following example. Let $X_1, X_2, \ldots, X_{10}$ be independent and identically distributed (iid) U(0, 1) random variables. Find

$$P\left(4 < \sum_{i=1}^{10} X_i < 6\right).$$

The distribution of the sum of 10 iid U(0, 1) random variables can be calculated by hand, but the process would be both tedious and time consuming. APPL solves this problem with the commands

```
> X := UniformRV(0, 1);
> Y := ConvolutionIID(X, 10);
> CDF(Y, 6) - CDF(Y, 4);
```

which yield the exact solution

$$\frac{655177}{907200}.$$

Although the solution to this problem can be approximated by the central limit theorem or Monte Carlo simulation, APPL computes the exact solution in just three lines of code.

APPL contains a univariate transformation of variables procedure that takes the probability density function (PDF) of a continuous random variable X and the transformation $Y = g(X)$ as input and returns the PDF of Y as output. APPL currently lacks procedures to handle bivariate distributions. The most relevant software we found to automate bivariate transformations is Mathematica's statistical package, mathStatica (Rose and Smith 2002). Their examples appear to be limited to one–to–one transformations and independent random variables X and Y. We present an addendum to APPL that handles one–to–one transformations, k–to–one transformations, and piecewise k–to–one transformations for both dependent and independent continuous random variables X and Y.

The bivariate transformation technique calculates the distribution of $U = g(X, Y)$ from the joint distribution of two continuous random variables X and Y with joint PDF $f_{X,Y}(x, y)$ defined on the support set $\mathscr{A}$ in the xy-plane. In the simplest case, an auxiliary dummy function $V = h(X, Y)$ can be found so that the two functions $U = g(X, Y)$ and $V = h(X, Y)$ define a one–to–one transformation from the set $\mathscr{A}$ in the xy-plane onto a set $\mathscr{B}$ in the uv-plane. Denote the inverse transformation from $\mathscr{B}$ to $\mathscr{A}$ by $X = r(U, V)$ and $Y = s(U, V)$. The formula (Hogg et al. 2005)

$$f_{U,V}(u, v) = f_{X,Y}(r(u, v), s(u, v))|J| \quad \text{for } (u, v) \in \mathscr{B}$$

gives the joint PDF of U and V, where

$$J = \begin{vmatrix} \dfrac{dx}{du} & \dfrac{dx}{dv} \\ \dfrac{dy}{du} & \dfrac{dy}{dv} \end{vmatrix}$$

is the Jacobian. The marginal PDF of U is then calculated by integrating v out of the joint PDF

$$f_U(u)=\int_{-\infty}^{\infty} f_{U,V}(u,v)\,dv=\int_{\mathcal{D}(u)} f_{U,V}(u,v)\,dv \quad \text{for } u\in\mathcal{U},$$

where $\mathcal{D}(u)=\{v \mid (u,v)\in\mathcal{B}\}$ and $\mathcal{U}$ is the support of U.

Next, we present a bivariate transformation algorithm used to handle such one–to–one transformations, and, more generally, k–to–one transformations and piecewise k–to–one transformations for the joint continuous random variables X and Y. The algorithm is modeled after the theorem by Glen et al. (1997) for univariate transformations.

2. Algorithm

The purpose of the algorithm is to find the PDF of $U=g(X,Y)$, where X and Y are bivariate continuous random variables with joint PDF $f_{X,Y}(x,y)$. Let the support of X and Y in the xy-plane be denoted by $\mathcal{A}$, which is partitioned into a finite number of disjoint components. Denote the ith component of $\mathcal{A}$ by $\mathcal{A}_i$ and the associated PDF on $\mathcal{A}_i$ by $f_i(x,y)$ for $i=1,2,\ldots,n$. The purpose of this partitioning might be to accommodate a distribution that is defined piecewise, a transformation that is defined piecewise, a transformation that is many-to-one, or any combination of these situations. Because the $\mathcal{A}_i$ might include some of their boundary points on which the transformation could be badly behaved, we consider the interior of each $\mathcal{A}_i$, denoted by $\mathcal{A}_i^\circ$, which is an open set. Let $(U,V)=w(X,Y)=(g(X,Y),h(X,Y))$ be a function from $\mathbb{R}^2$ to $\mathbb{R}^2$ whose domain includes $\bigcup_{i=1}^n \mathcal{A}_i^\circ$. The following assumptions must hold:

1. The PDF $f_i(x,y)$ is continuous for all $(x,y)\in\mathcal{A}_i^\circ$ for $i=1,2,\ldots,n$.
2. The function w is a one–to–one transformation from $\mathcal{A}_i^\circ$ onto a set $\mathcal{B}_i$ in the uv-plane, for $i=1,2,\ldots,n$. Denoting the inverse of this transformation from $\mathcal{A}_i^\circ$ onto $\mathcal{B}_i$ by $(X,Y)=w_i^{-1}(U,V)=(r_i(U,V),s_i(U,V))$, we require that the Jacobian

$$J_i=\begin{vmatrix} \dfrac{\partial r_i(u,v)}{\partial u} & \dfrac{\partial r_i(u,v)}{\partial v}\\[2mm] \dfrac{\partial s_i(u,v)}{\partial u} & \dfrac{\partial s_i(u,v)}{\partial v}\end{vmatrix}$$

is nonvanishing on $\mathcal{B}_i$ for $i=1,2,\ldots,n$ and that the partial derivatives are everywhere defined and continuous.

Because the $\mathcal{B}_i$s are not necessarily pairwise disjoint, we find the contribution to the joint PDF of U and V over the component $\mathcal{B}_i$ that results from $\mathcal{A}_i^\circ$, which is (Hogg et al. 2005)

$$f_{i,U,V}(u,v)=f_i(r_i(u,v),s_i(u,v))\,|J_i| \quad \text{for } (u,v)\in\mathcal{B}_i$$

for $i=1,2,\ldots,n$. Then, $f_{U,V}(u,v)=\sum f_{i,U,V}(u,v)$, where the sum is taken over all those is for which $(u,v)\in\mathcal{B}_i$. To calculate the marginal PDF $f_U(u)$, we define the following additional notation. For those values of u that can occur for points (u,v) in $\mathcal{B}_i$, denote the contribution of the component $\mathcal{A}_i$ to the marginal PDF of U by

$$f_{U_i}(u)=\int_{-\infty}^{\infty} f_{i,U,V}(u,v)\,dv=\int_{\mathcal{D}_i(u)} f_{i,U,V}(u,v)\,dv,$$

where $\mathcal{D}_i(u)=\{v \mid (u,v)\in\mathcal{B}_i\}$. To automate the integration over $\mathcal{D}_i(u)$, we have made the additional assumption that, for each value of u realizable in the interior of $\mathcal{B}_i$, $\mathcal{D}_i(u)$ is a single line segment (that is, each vertical line segment spanning $\mathcal{B}_i$ is entirely contained in $\mathcal{B}_i$) and that the end points of the closure of $\mathcal{D}_i(u)$ are determined by two different constraints. For some transformations defined on $\mathcal{A}$, these assumptions can only be satisfied by a judicious partitioning of $\mathcal{A}$. Because the boundary of $\mathcal{B}_i$ typically consists of many distinct constraint curves, and the limits of integration for v depend on which constraint curves are relevant for a particular value of u, the function $f_{U_i}(u)$ is defined in a piecewise fashion with m_i pieces, where m_i is a positive integer. Then proceed as follows to find the PDF of U.

- For $j=1,2,\ldots,m_i$, denote the jth piece of $f_{U_i}(u)$ by $f_{U_{ij}}(u)$, which is defined on an interval denoted by $u_{ij}<u<u_{i(j+1)}$.
- Let $U^*=\bigcup_{i=1}^n\{u_{ij} \mid j=1,2,\ldots,m_i+1\}$.
- Order the elements of U^* without repeats and relabel them using the notation u_i^* so that $u_1^*<u_2^*<\cdots<u_{l+1}^*$, where $l=\|U^*\|-1$, and $\|\cdot\|$ denotes cardinality.
- Let $I_k=\{(i,j) \mid u_{ij}\le u_k^* \text{ and } u_{k+1}^*\le u_{i(j+1)}\}$ for $k=1,2,\ldots,l$.

Then for $u\in(u_k^*,u_{k+1}^*)$, the PDF of U is given by

$$f_U(u)=\sum_{(i,j)\in I_k} f_{U_{ij}}(u)$$

for $k=1,2,\ldots,l$.

3. Data Structure

To implement the algorithm, we will use a data structure for the distribution of the bivariate random variable that expands on the list-of-sublists format used in APPL. The distribution of the bivariate random variable is presented in a list-of-three-sublists format. The first sublist contains the ordered two-variable PDF expressions $f_i(x,y)$ of the distribution, corresponding to the components $\mathcal{A}_i$ into which the support of X and Y has been partitioned. The second sublist describes each component of $\mathscr{A}_i$ by using a list of constraints for that component. The third sublist contains the strings `"Continuous"` and `"PDF"` to specify the type of distribution and meaning of the function in the first sublist. For proper implementation of our algorithms, each component described in the second sublist must be a simply connected set (i.e., contains no holes) whose boundary can be formed by consecutive segments of curves that satisfy equations of the form $p(x, y) = 0$. In addition, the constraints listed in the second sublist for each component must satisfy the following conditions:

- The constraints must be entered in adjacent order; clockwise or counterclockwise is acceptable.
- The constraints must completely enclose a region.
- The constraints must be entered as strict inequalities.
- Except for constraints of the form $x < a$ or $x > a$, each constraint, when written as an equality, must pass the vertical line test; i.e., only a single value of y corresponds to each value of x. For example, use $y < \sqrt{1-x^2}$ or $y > -\sqrt{1-x^2}$, rather than $x^2 + y^2 < 1$.
- If the component is unbounded, a less than ∞ or greater than $-\infty$ constraint must be included.

See Online Supplement A (available at http://joc.journal.informs.org/) for further discussion of this data structure and for examples illustrating its use.

4. Implementation

The bivariate transformation procedure `BiTransform(Z, g, h)` takes up to three arguments as input. The first two arguments are required and the third is optional. The first required argument is the joint distribution of X and Y, which is given in the list-of-three-sublists format. If $U = g(X, Y)$ is a k–to–one transformation or defined in a piecewise fashion, or both, then the user must partition the support of X and Y such that $g(X, Y)$ is one–to–one and is not defined in a piecewise fashion on each component of the partition. The second required argument is the transformation of interest $U = g(X, Y)$, given in the Maple function format `[(x, y)→ g(x, y)]`. If only one transformation is provided, then `g(x, y)` is applied to all components of the support of X and Y. Otherwise, the user must supply a number of transformations equal to the number of components, where the first transformation corresponds to the first component, the second transformation corresponds to the second component, and so on. The third optional argument is the transformation $V = h(X, Y)$, which is described using the same Maple function format and conventions used when describing U. The default transformation is the dummy transformation $V = h(X, Y) = Y$ for all values in $\mathscr{A}$.

We give detail in this paragraph concerning the automatic determination of the inverse. Each component $\mathscr{A}_i$ of the support of X and Y is associated with a corresponding transformation $U = g_i(X, Y)$ and $V = h_i(X, Y)$ for $i = 1, 2, \ldots, n$. The algorithm adapts to the case in which the inverse transformation $(X, Y) = (r_i(U, V), s_i(U, V))$ found by the Maple `solve` procedure returns multiple solutions, and the correct inverse must be chosen. For example, the two–to–one function $(U, V) = (X^2, Y)$ yields inverses $(X, Y) = (\sqrt{U}, V)$ and $(X, Y) = (-\sqrt{U}, V)$. The `BiTransform` procedure finds the correct inverse by selecting the inverse that satisfies $x_i = r_i(g_i(x_i, y_i), h_i(x_i, y_i))$ and $y_i = s_i(g_i(x_i, y_i), h_i(x_i, y_i))$ where $(x_i, y_i) \in \mathscr{A}_i^\circ$. An algorithm for determining such a point (x_i, y_i) in $\mathscr{A}_i$ follows. Let W_i denote the set of x-values for all intersections of adjacent constraint equalities that define $\mathscr{A}_i$. Let t_i be the minimum of W_i, and let T_i be the maximum of W_i. If W_i contains more than two elements, let $\hat{t}_i$ be the second-smallest element of W_i, and let $\widehat{T}_i$ be the second-largest element of W_i.

1. If $t_i = -\infty$ and $T_i = \infty$, then $x_i = 0$ and y_i is the average of the y-values for the two constraint equalities associated with $x_i = 0$ that maximize the difference of the y-values.
2. If $t_i = -\infty$ and T_i is finite, then $x_i = (T_i + \widehat{T}_i)/2$ and y_i is the average of the two y-values that correspond to x_i on the two constraint equalities associated with x_i.
3. If t_i is finite, then $x_i = (t_i + \hat{t}_i)/2$ and y_i is the average of the two y-values that correspond to x_i on the two constraint equalities associated with x_i.

Each constraint that defines $\mathscr{A}_i$ is an inequality of the form $p(x, y) < 0$, where p is a real-valued continuous function. The corresponding constraint for $\mathscr{B}_i$ is found by substituting the appropriate inverse transformations determined by the algorithm described in the previous paragraph to get $p(r_i(u, v), s_i(u, v)) < 0$. One difficulty arises when the number of constraints for $\mathscr{A}_i$ is greater than the number of constraints for $\mathscr{B}_i$. In other words, at least one of the corresponding constraints for $\mathscr{B}_i$ has been deleted as redundant. The following example illustrates this problem.

Consider the bivariate random variables X and Y with support on the open unit square $0 < x < 1$, $0 < y < 1$. The transformation $U = XY$ with dummy transformation $V = Y$ is a one–to–one transformation from $\mathscr{A}$ to $\mathscr{B}$ (shown in Figure 1 where the boundaries of $\mathscr{A}$ and $\mathscr{B}$ are not a part of the support) with inverse $X = U/V$, $Y = V$. Substituting $X = U/V$ and $Y = V$ into the linear constraints $x > 0$, $y > 0$, $x < 1$, $y < 1$ gives the linear constraints $u > 0$, $v > 0$, $u < v$, $v < 1$. The second constraint, $v > 0$, is redundant and

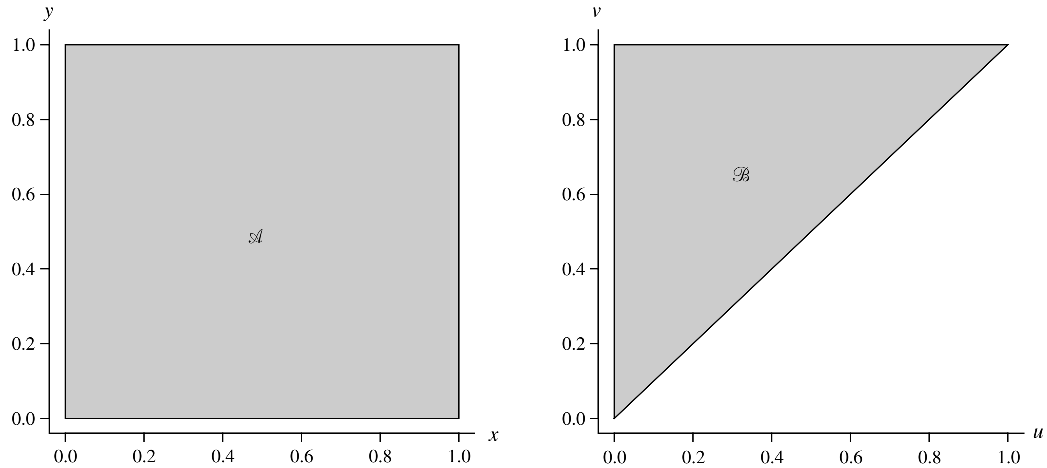

Figure 1 **Transformation from $\mathscr{A}$ to $\mathscr{B}$ Given by** $(U, V) = (XY, Y)$
Note. The constraint equality $y = 0$ of $\mathscr{A}$ is mapped to a point on the boundary of $\mathscr{B}$.

should not be included in the description of the support of UV.

The support of X and Y is bounded by four linear constraints whose corresponding equalities intersect at the adjacent points $(0,0)$, $(1,0)$, $(1,1)$, $(0,1)$. These points on the boundary of $\mathscr{A}$ are mapped to $(0,0)$, $(0,0)$, $(1,1)$, $(0,1)$, respectively, on the boundary of $\mathscr{B}$. The `BiTransform` procedure checks for equality between adjacent intersections for $\mathscr{B}$ and deletes redundant constraints. In this case, the first and second intersections for $\mathscr{A}$ map to the same point, $(0,0)$, for $\mathscr{B}$, so $v > 0$ is deleted.

There are cases where Maple functions do not operate as desired. The most troublesome of these functions is `solve`. In particular, the input `solve({x = a, y = infinity})` returns `NULL` for $a = 0$ but returns `{x = a, y =`∞`}` for all $a \neq 0$. The `solve` procedure is used primarily to find the intersection points between adjacent constraints associated with the support. Because joint distributions constrained to the first quadrant are fairly common, "if" statements in the code handle the $a = 0$ case. However, it is impossible to add conditional statements for all problems of this type (e.g., `solve({x = y * y, y = infinity})` also returns `NULL`). Thus, the capability of our transformation procedure, `BiTransform`, is limited in handling distributions with infinite support. The scope of problems that can be solved by `BiTransform` is limited to those whose adjacent constraint equalities can be solved analytically by the `solve` function.

5. Examples

This section illustrates the use of the bivariate transformation procedure `BiTransform` for a variety of distributions and transformations. Example 1 illustrates the use of `BiTransform` on independent random variables. Example 2 illustrates transforming dependent random variables. Example 3 illustrates piecewise transformations. Example 4 illustrates k–to–one transformations. Example 5 illustrates piecewise k–to–one transformations. Example 6 illustrates a method for working around a support that is not a simply connected set.

Example 1 (Independent X and Y). The joint PDF of the independent continuous random variables $X \sim \mathrm{U}(0,1)$ and $Y \sim \mathrm{U}(0,1)$ is

$$f_{X,Y}(x,y) = f_X(x) f_Y(y) = 1, \quad 0 < x < 1,\ 0 < y < 1.$$

The distribution of $U = XY$ is found using the transformation technique as follows:

$$u = g(x,y) = xy, \qquad x = r(u,v) = u/v,$$
$$v = h(x,y) = y, \qquad y = s(u,v) = v,$$
$$J = \begin{vmatrix} 1/v & -u/v^2 \\ 0 & 1 \end{vmatrix} = 1/v,$$
$$f_{U,V}(u,v) = 1 \cdot \left|\frac{1}{v}\right| = \frac{1}{v}, \quad 0 < u < v,$$
$$f_U(u) = \int_u^1 \frac{1}{v}\,dv = [\ln v]_u^1 = -\ln u, \quad 0 < u < 1.$$

The commands to assign data structures and apply the bivariate transformation technique are as follows:

```
> XY := [[(x, y) -> 1], [[y > 0, x > 0, y < 1,
              x < 1]], ["Continuous", "PDF"]];
> g := [(x, y) -> x * y];
> h := [(x, y) -> y];
> U := BiTransform(XY, g, h);
```

which return

$$[[u \to -\ln(u)], [0, 1], [\text{“}Continuous\text{”}, \text{“}PDF\text{”}]]$$

as the PDF of U as expected. Leaving out the third line of code and the third argument in `BiTransform` results in identical output.

Example 2 (Dependent X and Y). The joint PDF of dependent continuous random variables X and Y is

$$f_{X,Y}(x, y) = 2, \quad x > 0,\ y > 0,\ x + y < 1.$$

The distribution of $U = X + Y$ is found using the transformation technique as follows:

$$u = g(x, y) = x + y, \qquad x = r(u, v) = \frac{u+v}{2},$$

$$v = h(x, y) = x - y, \qquad y = s(u, v) = \frac{u-v}{2},$$

$$J = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{vmatrix} = -\frac{1}{2},$$

$$f_{U,V}(u, v) = 2 \cdot \left| -\frac{1}{2} \right| = 1, \quad 0 < u < 1,\ -u < v < u,$$

$$f_U(u) = \int_{-u}^{u} 1\, dv = 2u, \quad 0 < u < 1.$$

The commands to assign data structures and apply the bivariate transformation technique are as follows:

```
> XY := [[(x, y) -> 2], [[x > 0, y > 0,
          x + y < 1]], ["Continuous", "PDF"]];
> g := [(x, y) -> x + y];
> h := [(x, y) -> x - y];
> U := BiTransform(XY, g, h);
```

which correctly return

$$[[u \to 2\,u], [0, 1], [\text{“}Continuous\text{”}, \text{“}PDF\text{”}]]$$

as the PDF of U.

Example 3 (Piecewise Transformation). The joint PDF of the independent continuous random variables $X \sim \mathrm{U}(0, 1)$ and $Y \sim \mathrm{U}(0, 1)$ is

$$f_{X,Y}(x, y) = f_X(x) f_Y(y) = 1, \quad 0 < x < 1,\ 0 < y < 1.$$

Let U be a piecewise transformation with $U = X + Y$ for $0 < x < 1/2$, $0 < y < 1$, and $U = XY$ for $1/2 < x < 1$, $0 < y < 1$. The transformation is illustrated in Figure 2, using $V = X - Y$ on $0 < X < 1/2$, $0 < Y < 1$ and $V = Y$ on $1/2 < X < 1$, $0 < Y < 1$. For $\mathcal{A}_1$ with $0 < x < 1/2$, $0 < y < 1$,

$$u = g_1(x, y) = x + y, \qquad x = r_1(u, v) = \frac{u+v}{2},$$

$$v = h_1(x, y) = x - y, \qquad y = s_1(u, v) = \frac{u-v}{2},$$

$$J_1 = \begin{vmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{vmatrix} = -\frac{1}{2},$$

$$f_{1,U,V}(u, v) = 1 \cdot \left| -\frac{1}{2} \right| = 1/2,\ 0 < u + v < 1,\ 0 < u - v < 2,$$

$$f_{U_1}(u) = \begin{cases} \int_{-u}^{u} 1/2\, dv & 0 < u < 1/2, \\ \int_{-u}^{1-u} 1/2\, dv & 1/2 < u < 1, \\ \int_{u-2}^{1-u} 1/2\, dv & 1 < u < 3/2, \end{cases}$$

$$= \begin{cases} u & 0 < u < 1/2, \\ 1/2 & 1/2 < u < 1, \\ (3 - 2u)/2 & 1 < u < 3/2. \end{cases}$$

For $\mathcal{A}_2$ with $u = g_2(x, y) = xy$, $v = h_2(x, y) = y$, we have $J_2 = 1/v$ from Example 1 and

$$f_{2,U,V}(u, v) = 1 \cdot \left| \frac{1}{v} \right| = \frac{1}{v}, \quad v/2 < u < v,\ 0 < v < 1,$$

$$f_{U_2}(u) = \begin{cases} \int_{u}^{2u} 1/v\, dv = \ln(2) & 0 < u < 1/2, \\ \int_{u}^{1} 1/v\, dv = -\ln(u) & 1/2 < u < 1. \end{cases}$$

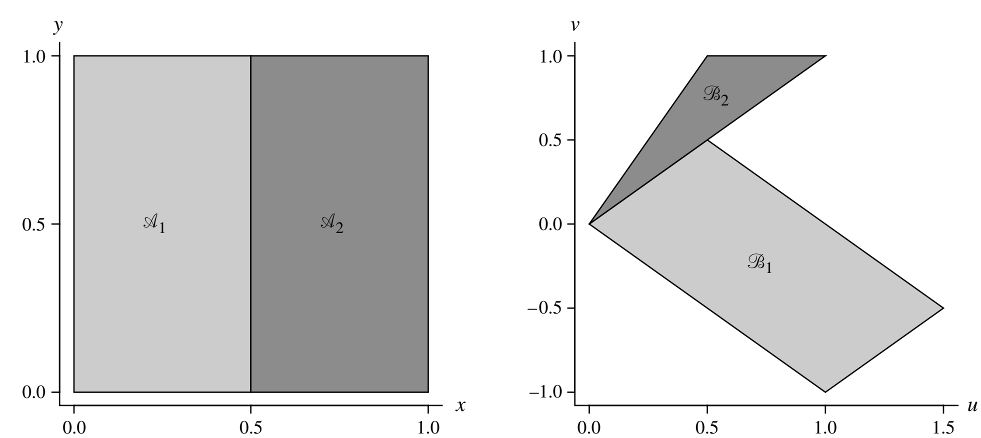

Figure 2 **Piecewise Transformation from $\mathcal{A}_1$ to $\mathcal{B}_1$ and $\mathcal{A}_2$ to $\mathcal{B}_2$**

Using the algorithm for bivariate transformations,

$$f_U(u)=f_{U_1}(u)+f_{U_2}(u)=\begin{cases} u+\ln(2) & 0<u<1/2,\\ 1/2-\ln(u) & 1/2<u<1,\\ (3-2u)/2 & 1<u<3/2.\end{cases}$$

The commands to assign data structures and apply the bivariate transformation technique are as follows:

```
> XY := [[(x, y) -> 1, (x, y) -> 1],
         [[y > 0, x < 0, y < 1, x < 1/2],
         [y > 0, x > 1/2, y < 1, x < 1]],
         ["Continuous", "PDF"]];
> g := [(x, y) -> x + y, (x, y)-> x * y];
> h := [(x, y) -> x - y, (x, y)-> y];
> U := BiTransform(XY, g, h);
```

which return the correct PDF of U in the APPL list-of-sublists format as

$$[[u\to u+\ln(2),\, u\to -\ln(u)+1/2,\, u\to -u+3/2],\\ [0, 1/2, 1, 3/2], [\text{“}Continuous\text{”}, \text{“}PDF\text{”}]].$$

Example 4 (Two–to–One Transformation). The joint PDF of the independent continuous random variables X and Y is

$$f_{X,Y}(x,y)=1/4,\quad -1<x<1,\ -1<y<1.$$

The distribution of U under the two–to–one transformation $U=X^2-Y$ and $V=Y$ is found using the algorithm described in §2. Partition the support of X and Y such that the transformation is one–to–one in each component. Let $\mathscr{A}_1$ be the component bound by $-1<x<0,\ -1<y<1$; and let $\mathscr{A}_2$ be the component bound by $0<x<1,\ -1<y<1$. Both components map onto $\mathscr{B}$ under the transformation $U=X^2-Y$ and $V=Y$, as illustrated in Figure 3.

For $\mathscr{A}_1$,

$$u=g_1(x,y)=x^2-y,\qquad x=r_1(u,v)=-(u+v)^{1/2},$$
$$v=h_1(x,y)=y,\qquad y=s_1(u,v)=v,$$
$$J_1=\begin{vmatrix}-\frac{1}{2}(u+v)^{-1/2} & -\frac{1}{2}(u+v)^{-1/2}\\ 0 & 1\end{vmatrix}=-\tfrac{1}{2}(u+v)^{-1/2},$$
$$f_{1,U,V}(u,v)=\tfrac{1}{4}\cdot\left|-\tfrac{1}{2}(u+v)^{-1/2}\right|=\tfrac{1}{8}(u+v)^{-1/2},$$
$$-v<u<1-v,\ -1<v<1,$$

$$f_{U_1}(u)=\begin{cases}\int_{-u}^{1}\frac{1}{8}(u+v)^{-1/2}\,dv & -1<u<0,\\ \int_{-u}^{1-u}\frac{1}{8}(u+v)^{-1/2}\,dv & 0<u<1,\\ \int_{-1}^{1-u}\frac{1}{8}(u+v)^{-1/2}\,dv & 1<u<2,\end{cases}$$
$$=\begin{cases}\sqrt{u+1}/4 & -1<u<0,\\ 1/4 & 0<u<1,\\ -\sqrt{u-1}/4+1/4 & 1<u<2.\end{cases}$$

For $\mathscr{A}_2$,

$$u=g_2(x,y)=x^2-y,\qquad x=r_2(u,v)=(u+v)^{1/2},$$
$$v=h_2(x,y)=y,\qquad y=s_2(u,v)=v,$$
$$J_2=\begin{vmatrix}\frac{1}{2}(u+v)^{-1/2} & \frac{1}{2}(u+v)^{-1/2}\\ 0 & 1\end{vmatrix}=\tfrac{1}{2}(u+v)^{-1/2},$$
$$f_{2,U,V}(u,v)=\tfrac{1}{4}\cdot\left|\tfrac{1}{2}(u+v)^{-1/2}\right|=\tfrac{1}{8}(u+v)^{-1/2},$$
$$-v<u<1-v,\ -1<v<1,$$

$$f_{U_2}(u)=\begin{cases}\int_{-u}^{1}\frac{1}{8}(u+v)^{-1/2}\,dv & -1<u<0,\\ \int_{-u}^{1-u}\frac{1}{8}(u+v)^{-1/2}\,dv & 0<u<1,\\ \int_{-1}^{1-u}\frac{1}{8}(u+v)^{-1/2}\,dv & 1<u<2,\end{cases}$$

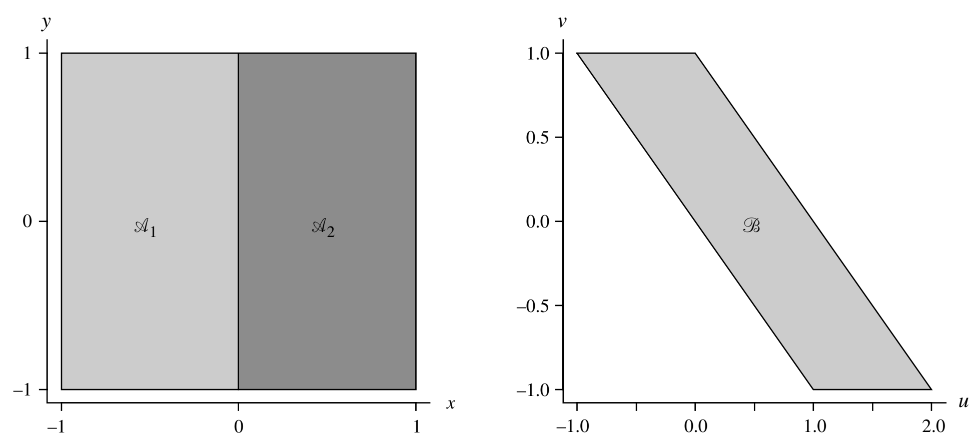

Figure 3 **Two–to–One Transformation from $\mathscr{A}=\mathscr{A}_1\cup\mathscr{A}_2$ to $\mathscr{B}$**

$$= \begin{cases} \sqrt{u+1}/4 & -1 < u < 0, \\ 1/4 & 0 < u < 1, \\ -\sqrt{u-1}/4 + 1/4 & 1 < u < 2. \end{cases}$$

Using the algorithm for bivariate transformations,

$$f_U(u) = f_{U_1}(u) + f_{U_2}(u)$$
$$= \begin{cases} \sqrt{u+1}/2 & -1 < u < 0, \\ 1/2 & 0 < u < 1, \\ -\sqrt{u-1}/2 + 1/2 & 1 < u < 2. \end{cases}$$

The commands to assign data structures and apply the bivariate transformation technique are as follows:

```
> XY := [[(x, y) -> 1/4, (x, y) -> 1/4],
        [[y > -1, x > -1, y < 1, x < 0],
        [y > -1, x > 0, y < 1, x < 1]],
        ["Continuous", "PDF"]];
> g := [(x, y) -> x ^ 2 - y];
> h := [(x, y) -> y];
> U := BiTransform(XY, g, h);
```

which return the correct PDF of U in the APPL list-of-sublists format as

$$[[u \to \sqrt{u+1}/2, u \to 1/2, u \to -\sqrt{u-1}/2 + 1/2], [-1, 0, 1, 2], [\text{“}Continuous\text{''}, \text{“}PDF\text{''}]].$$

Example 5 (Piecewise Two–to–One Transformation). The joint PDF of independent continuous random variables X and Y is

$$f_{X,Y}(x, y) = 1/6, \quad -2 < x < 1, -1 < y < 1.$$

The transformation $U = X^2 - Y$ and $V = Y$ is two–to–one on $-1 < x < 1$, $-1 < y < 1$ and one–to–one on $-2 < x < -1$, $-1 < y < 1$. Partition the support of X and Y such that the transformation $U = X^2 - Y$ and $V = Y$ is one–to–one in each component. Let $\mathcal{A}_1$ be the component bound by $-2 < x < 0$, $-1 < y < 1$; and let $\mathcal{A}_2$ be the component bound by $0 < x < 1$, $-1 < y < 1$, as illustrated in Figure 4.

The marginal PDF of U is

$$f_U(u) = \begin{cases} \sqrt{u+1}/3 & -1 < u < 0, \\ \sqrt{u+1}/6 + 1/6 & 0 < u < 1, \\ -\sqrt{u-1}/3 + 1/6 + \sqrt{u+1}/6 & 1 < u < 2, \\ -\sqrt{u-1}/6 + \sqrt{u+1}/6 & 2 < u < 3, \\ -\sqrt{u-1}/6 + 1/3 & 3 < u < 5. \end{cases}$$

The commands to assign data structures and apply the bivariate transformation technique are as follows:

```
> XY := [[(x, y) -> 1/6, (x, y) -> 1/6],
        [[y > -1, x > -2, y < 1, x < 0],
        [y > -1, x > 0, y < 1, x < 1]],
        ["Continuous", "PDF"]];
> g := [(x, y) -> x ^ 2 - y, (x, y) -> x ^ 2 - y];
> h := [(x, y) -> y, (x, y) -> y];
> U := BiTransform(XY, g, h);
```

Note that g and h each contain two functions instead of one as in the previous example. Declaring the same transformation for each component achieves the same result as declaring the transformation once. The correct PDF of U is returned in the APPL list-of-sublists format as

$$[[u \to \sqrt{u+1}/3, u \to \sqrt{u+1}/6 + 1/6, u \to -\sqrt{u-1}/3 + 1/6 + \sqrt{u+1}/6, u \to -\sqrt{u-1}/6 + \sqrt{u+1}/6, u \to -\sqrt{u-1}/6 + 1/3], [-1, 0, 1, 2, 3, 5], [\text{“}Continuous\text{''}, \text{“}PDF\text{''}]].$$

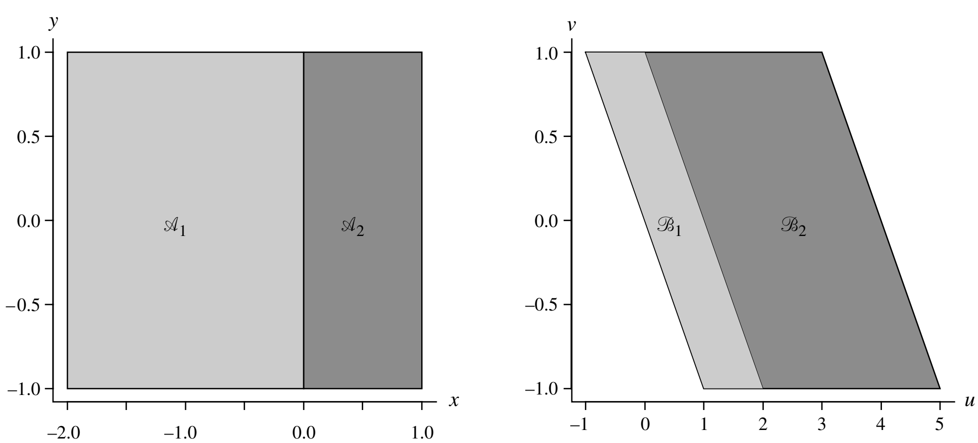

Figure 4 **Piecewise Two–to–One Transformation**
Note. The region $\mathcal{A}_1$ maps onto $\mathcal{B}_1 \cup \mathcal{B}_2$, whereas $\mathcal{A}_2$ only maps to $\mathcal{B}_2$.

Example 6 (Nonsimply Connected Support for U and V). Let X and Y be uniformly distributed on $(1,2)\times(-\pi,\pi)$ with transformations

$$U = X\cos Y, \qquad V = X\sin Y.$$

The joint density of U and V is

$$f_{U,V}(u,v) = \frac{1}{2\pi\sqrt{u^2+v^2}}, \quad 1 < u^2+v^2 < 4,$$

and the marginal PDF of U is

$$f_U(u) = \begin{cases} \pi^{-1}\sinh^{-1}\sqrt{4u^{-2}-1} & 1<|u|<2,\\ \pi^{-1}(\sinh^{-1}\sqrt{4u^{-2}-1} - \sinh^{-2}\sqrt{u^{-1}-1}) & \\ & |u|<1. \end{cases}$$

The commands to assign data structures and apply the bivariate transformation technique are as follows:

```
> X := UniformRV(1, 2);
> Y := UniformRV(-Pi, Pi);
> XY := JointPDF(X, Y);
> g := [(x, y) -> x * cos(y)];
> h := [(x, y) -> x * sin(y)];
> BiTransform(XY, g, h);
```

which displays the joint density of (U,V) as

$$\left[\left[(u,v)\to 1/2\frac{1}{\pi\sqrt{|v^2+u^2|}}\right],\right.$$

$$\left[\left[1<\sqrt{v^2+u^2}, \sqrt{v^2+u^2}<2,\right.\right.$$

$$\left.\left.\arctan\left(\frac{v}{\sqrt{v^2+u^2}}, \frac{u}{\sqrt{v^2+u^2}}\right)<\pi\right]\right],$$

$$\left.[\text{“}Continuous\text{”}, \text{“}PDF\text{”}]\right].$$

The marginal PDF is calculated by manually dividing the support of (U,V) into six regions using $u=-1$, $v=0$, and $u=1$, as illustrated in Figure 5.

The Maple commands

```
> UV := [[(u, v) -> 1 / (2 * Pi * sqrt(u ^ 2 + v ^ 2)),
          (u, v) -> 1 / (2 * Pi * sqrt(u ^ 2 + v ^ 2)),
          (u, v) -> 1 / (2 * Pi * sqrt(u ^ 2 + v ^ 2)),
          (u, v) -> 1 / (2 * Pi * sqrt(u ^ 2 + v ^ 2)),
          (u, v) -> 1 / (2 * Pi * sqrt(u ^ 2 + v ^ 2)),
          (u, v) -> 1 / (2 * Pi * sqrt(u ^ 2 + v ^ 2))],
        [[v < sqrt(4 - u ^ 2), u < -1, v > 0],
         [v < sqrt(4 - u ^ 2), u > -1,
          v > sqrt(1 - u ^ 2), u < 1],
         [v < sqrt(4 - u ^ 2), u > 1, v > 0],
         [v > -sqrt(4 - u ^ 2), u < -1, v < 0],
         [v > -sqrt(4 - u ^ 2), u > -1,
          v < -sqrt(1 - u ^ 2), u < 1],
         [v > -sqrt(4 - u ^ 2), u > 1, v < 0]],
         ["Continuous", "PDF"]];
> MarginalPDF(UV, u);
```

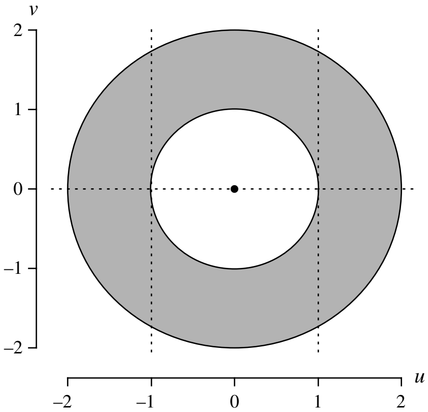

Figure 5 Support of U and V

correctly return

$$\left[\left[u\to -\frac{-1/2\ln(\sqrt{-u^2+4}+2)+1/2\ln(-\sqrt{-u^2+4}+2)}{\pi},\right.\right.$$

$$u\to \frac{1}{\pi}\cdot(-1/2\ln(\sqrt{-u^2+1}+1)-1/2\ln(\sqrt{-u^2+4}+2)$$

$$+1/2\ln(-\sqrt{-u^2+4}+2)-1/2\ln(-\sqrt{-u^2+1}+1)),$$

$$\left.u\to -\frac{-1/2\ln(\sqrt{-u^2+4}+2)+1/2\ln(-\sqrt{-u^2+4}+2)}{\pi}\right],$$

$$\left.[-2.0,-1.0,1.0,2.0],[\text{“}Continuous\text{”}, \text{“}PDF\text{”}]\right]$$

as the marginal PDF of U. [Recall that $\sinh^{-1}(x) = \ln(x+\sqrt{x^2+1})$.]

6. Conclusions and Further Research

Calculating the distribution of a transformation of two random variables by hand can be tedious, especially when working with piecewise joint distributions or piecewise transformations. An automated method for computing bivariate transformations saves the user time and prevents calculation errors. The algorithm and `BiTransform` procedure we developed allows for one–to–one, k–to–one, and piecewise k–to–one transformations for both independent and dependent continuous random variables. In the future, we hope to see computer-based algorithms for bivariate discrete transformations, as well as multivariate transformations involving three or more random variables. There are numerous applications of `BiTransform` in stochastic operations research (one from queuing theory is given in Online Supplement F). The algorithm described in this paper has been implemented in approximately 1,300 lines of code and is available at http://www.math.wm.edu/~leemis/BiVarAPPL.txt. The APPL code can be downloaded at http://www.APPLsoftware.com.

Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at http://joc.journal.informs.org/.

Acknowledgments

The authors thank Area Editor Winfried Grassmann, an associate editor, and two referees for their helpful comments and for suggesting two examples that are included in the paper.

References

Glen, A. G., D. L. Evans, L. M. Leemis. 2001. APPL: A probability programming language. *Amer. Statistician* **55**(2) 156–166.

Glen, A. G., L. M. Leemis, J. H. Drew. 1997. A generalized univariate change-of-variable transformation technique. *INFORMS J. Comput.* **9**(3) 288–295.

Hogg, R. V., J. W. McKean, A. T. Craig. 2005. *Introduction to Mathematical Statistics*, 6th ed. Prentice Hall, Upper Saddle River, NJ.

Rose, C., M. D. Smith. 2002. *Mathematical Statistics with Mathematica*. Springer-Verlag, New York.