fall within the region $\mathcal{R}$

$$V_{\mathcal{R}} = \frac{N_{\mathcal{R}}}{N} L^n.$$  \hfill (14.2)

The integral formula (14.1) then becomes

$$\int_{\mathcal{R}} f(x) \, d^n x \approx \frac{L^n}{N} \sum_{k=1}^{N_{\mathcal{R}}} f(x_k).$$  \hfill (14.3)

The utility of the Monte Carlo method of numerical integration rises sharply with the dimension $n$ of the hypervolume.

**Example 14.1** (Numerical Integration) Suppose one wants to integrate the function

$$f(x, y) = \frac{e^{-2x - 3y}}{\sqrt{x^2 + y^2 + 1}}$$  \hfill (14.4)

over the quarter of the unit disk in which $x$ and $y$ are positive. In this case, $V_{\mathcal{R}}$ is the area $\pi/4$ of the quarter disk.

To generate fresh random numbers, one must set the seed for the code that computes them. The following program sets the seed by using the subroutine `init_random_seed` defined in a fortran95 program in section 13.16. With some compilers, one can just write “call random_seed().”

```fortran
program integrate
  implicit none ! catches typos
  integer :: k, N
  integer, parameter :: dp = kind(1.0d0)
  real(dp) :: x, y, sum = 0.0d0, f
  real(dp), dimension(2) :: rdn
  real(dp), parameter :: area = atan(1.0d0) ! pi/4
  f(x,y) = exp(-2*x - 3*y)/sqrt(x**2 + y**2 + 1.0d0)
  write(6,*) 'How many points?'
  read(5,*) N
  call init_random_seed() ! set new seed
  do k = 1, N
    call random_number(rdn); x= rdn(1); y = rdn(2)
    if (x**2+y**2 > 1.0d0) then
      go to 10
    end if
    sum = sum + f(x,y)
  end do
  ! integral = area times mean value < f > of f
```