

NAME

Fmdn() – Virtual waiting time distribution function
 IntFmdn() – Virtual waiting time distribution function for integral values of x
 Mdn() – Calculates the state probabilities of a queue (queue length)
 Fekdn() – Virtual waiting time distribution function

SYNOPSIS

```
#include <queue.h>

double Fmdn(double x, double rho, int n);
double IntFmdn(int N, double rho, int n);
double *Mdn(double x, double rho, int n);
double Fekdn(double x, double rho, int k, int n);
```

DESCRIPTION

delim \$\$ These functions return the state probabilities or the virtual waiting time distribution of a $M/D/n$ queue. (And the virtual waiting time distribution of a $E_k/D/n$ queue.)

Fmdn() is a model for the $M/D/n$ queueing system with Poisson arrivals and deterministic (constant) service time. Parameter x is the amount of unfinished work in the system. Rho is the load level of the system and n is the number of servers.

Fekdn() is a model for the $E_k/D/n$ queueing system with Erlang- k arrivals and deterministic (constant) service time. Parameter x is the amount of unfinished work in the system. Rho is the load level of the system and n is the number of servers.

ALGORITHM

$M/D/n$ waiting time distribution is calculated using the following algorithm (Iversen):

$$P\{SW \leq t\} = \sum_{i=0}^{n-1} \sum_{j=0}^i P(j) \sum_{\nu=0}^{\{T\} - \{A(\nu)-t\}} \{A(\nu)-t\} \sup_{\nu+n+\nu-1-i} \{A(\nu)-t\} e^{\sup_{\nu+n+\nu-1-i} \{A(\nu)-t\}}$$

where $P(j)$ is a state probability.

For integral values of the waiting time we have

$$P\{SW \leq t\} = \sum_{\nu=0}^{n(t+1)-1} P(\nu).$$

The state probabilities are calculated using the following procedure

– first we make an initial guess $(M/M/n)$:

$$\begin{aligned} P(0) &= \left(\sum_{i=0}^{n-1} \frac{(\lambda h)^i}{i!} + \frac{(\lambda h)^n}{n!} \right) \sup_n \{1 - \lambda h\} \\ P(i) &= P(i-1) \frac{\lambda h}{i}, \quad i=1,2,\dots,n-1 \\ P(i) &= P(i-1) \frac{\lambda h}{n}, \quad i=n,n+1,\dots,I \text{ (until } P(i) < \epsilon) \\ P(i) &= P(i-1) \frac{\lambda h}{n}, \quad i=I+1,\dots,I+n \end{aligned}$$

– then we iterate until $\max_{i \leq I} |P(k)(i) - P(k-1)(i)| < \epsilon$:

$$\begin{aligned} P(k)(i) &= \sum_{\nu=0}^n P(k-1)(\nu) P(i, h) + \sum_{\nu=n+1}^{n+i} P(k-1)(\nu) P(n+i-\nu, h), \quad i=0,1,\dots,I \\ P(k)(i) &= P(k)(i-1) \frac{\lambda h}{n}, \quad i=I+1,\dots,I+n \\ S(k) &= \sum_{i=0}^{I+n} P(k)(i) \\ P(k)(i) &= P(k)(i) \sup_{S(k)} \{P(k)(i)\}, \quad i=0,1,\dots,I+n \\ P(i, h) &= \left(\frac{(\lambda h)^i}{i!} + \frac{(\lambda h)^n}{n!} \right) e^{-\lambda h} \end{aligned}$$

$E_k/D/n$ waiting time distribution is calculated using the $M/D/n$ algorithm:

$E_k/D/r$ (FIFO) is equivalent to $M/D/r*k$ (FIFO).

ERRORS

When ρ is close to 1, these functions might give inaccurate results.

SEE ALSO

COST 224: Performance evaluation and design of multiservice networks