

**NAME**

Fmd1() – Virtual waiting time distribution function

IntFmd1() – Virtual waiting time distribution function for integral values of  $x$

SumMd1() – Calculates the state probabilities of a M/D/1 queue

RecMd1() – Calculates the state probabilities of a M/D/1 queue using a recursive algorithm

**SYNOPSIS**

```
#include <queue.h>
```

```
double Fmd1(double x, double rho);
```

```
double IntFmd1(int N, double rho);
```

```
double *SumMd1(double x, double rho);
```

```
double *RecMd1(int x, double rho);
```

**DESCRIPTION**

delim \$\$ These functions return the state probabilities or the virtual waiting time distribution of a M/D/1 queue.

**Fmd1()** is a model for the M/D/1 queuing 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.

**ALGORITHM**

M/D/1 waiting time distribution is calculated using the following algorithm (Iversen):

$$P\{W \leq T + \tau\} = e^{\lambda \tau} \sum_{0 \leq n \leq T} \{(-\lambda \tau)^n \sup_{n!}\} P\{W \leq T - n\},$$

where  $P\{W \leq T - n\}$  is the waiting time for integral values of  $x$ :

$$P\{W \leq t\} = P\{0\} + P\{1\} + \dots + P\{t\}.$$

The state probabilities can be calculated by using a general or a recursive algorithm (Fmd1() uses the recursive algorithm, because it is more accurate):

$$\begin{aligned} P(0) &= 1 - A \\ P(1) &= (1 - A)(e^A - 1) \\ P(2) &= (1 - A)(-e^A(1 + A) + e^{2A}) \\ A &= \{\lambda h\} \text{ (If } h=1, A=\{\lambda\} \text{).} \end{aligned}$$

General algorithm:

$$P(i) = (1 - A) \sum_{1 \leq n \leq i} (-1)^{n-1} e^{nA} \left( \sum_{n=1}^i \{(-nA)^{i-n}\} \sup_{(i-n)!}\right) + \left( \sum_{n=1}^{i-1} \{(-nA)^{i-n-1}\} \sup_{(i-n-1)!}\right), i=2,3,\dots$$

(The last term always equals  $\{e^{iA}\} \sup_{i!}$ .)

Recursive algorithm:

$$\begin{aligned} P(i+1) &= \{1 \sup_{P(0,h)}\} \{P(i) - [P(0) + P(1)] P(i,h) - \sum_{2 \leq n \leq i} P(n) P(i-n+1,h)\}, i=3,4,\dots \\ P(i,h) &= \left( \sum_{i=1}^i \{(\lambda h)^i\} \sup_{i!}\right) e^{-\{\lambda h\}} \end{aligned}$$

**ERRORS**

When  $\rho$  is close to 1, these functions might give inaccurate results.

**SEE ALSO**

COST 224: Performance evaluation and design of multiservice networks

