On the Asymptotic Behaviour of Closed Multiclass Queueing Networks

Peter G. Harrison Sérgio Coury

Department of Computing
Imperial College
180 Queen's Gate, London SW7 2BZ

Abstract

An expression for the generating function of the normalising constant is obtained for a closed, multiclass Markovian queueing network in terms of similar functions for networks with one fewer class. In this way a recursive algorithm can be implemented to calculate normalising constants, and hence many performance measures, in terms of the normalising constant function for single class networks, which is well understood. The technique is illustrated for a simple two-class network and explains simply some interesting asymptotic properties.

QUEUEING NETWORKS, MULTIPLE CLASSES, NORMALIZING CONSTANT, ASYMPTOTIC ANALYSIS, STOCHASTIC NETWORKS

1 Introduction

In order to obtain the normalising constant of a closed, product-form q-class queueing network, we seek the coefficient of $z_q^{k_q}$ in the normalising constant's generating function $G(z_1, ..., z_q)$, where k_q is the population of class q. This gives a result in terms of the vector $(z_1, ..., z_{q-1})$ which leads to a mixture of generating functions for normalising constants in networks with only $(q-1)$ classes. Repeating this process recursively results in an expression which is a mixture of single class network normalising constants. Although for large numbers of classes, the number of single class normalising constants that results may be considerable, for small numbers at least, the method is efficient. Moreover, it can explain directly certain interesting asymptotic properties, such as bottleneck behaviour with respect to class population.

There is a considerable literature on the subject of normalising constants, beginning with the single-class work of [Mitra and McKenna 1986] which was based on an integral representation of the normalising constant, obtained by using Euler's integral formula for the factorial function. This analysis is restricted to certain limits of what is called *network usage* — a parameter related to the node utilisation. Other asymptotic analyses include the work of [Knessl and Tier 1990], based on ray methods from geometric optics, and the fluid-flow-based approximation of [Birman et al. 1992], which does consider multiple classes.

In the next section, the main result is obtained, expressing the generating function of a normalising constant in terms of those for one less class. Section 3 then considers a two class network and the paper concludes in section 4.

2 Recursive formulation

In order to obtain the normalising constant of a multi-class network, we invert its generating function by evaluating a multiple integral over a complex multi-dimensional space. We then apply the Cauchy Residue Theorem (see [Guillemin 1949]) to obtain the normalising constant of a q-class network in terms of the normalising constants of smaller networks with $q-1$ classes. The method can then be repeated until a collection of single class networks is reached.

Consider a closed product form multi-class queueing network. Let q be the number of classes of jobs and X the number of service centers – also referred to as *nodes*. Let the q-dimensional quantity $\boldsymbol{\rho}_i = (\rho_{i1}, \ldots, \rho_{iq})$ be the node i load vector $(1 \leq i \leq X)$, where the load ρ_{ij} is the expected number of visits of class j jobs to node i in some given time period at equilibrium, divided by the service rate of class j jobs at node i. The *type* of a node is its load vector and the *multiplicity* of a type is the number of nodes in the network having that same load vector.

We consider a network with distinct node types 1 to M, of multiplicity $m_i \geq 1$ $(1 \leq i \leq M \leq X)$. It is straightforward to verify that the generating function of the normalising constant is then given by:

$$G(\mathbf{z}) = \sum_{k_1=0}^{\infty} \ldots \sum_{k_q=0}^{\infty} g(\mathbf{k}) z_1^{k_1} \ldots z_q^{k_q} = \prod_{i=1}^{M} \frac{1}{(1-\boldsymbol{\rho}_i \mathbf{z})^{m_i}} \tag{1}$$

where $\mathbf{z} = (z_1, \ldots, z_q)$, $\mathbf{k} = (k_1, \ldots, k_q)$ and $\boldsymbol{\rho}_i \mathbf{z} = \sum_{j=1}^{q} \rho_{ij} z_j$. The required normalising constant is $g(\mathbf{k})$ for a given class population vector $\mathbf{k}$.

Let $G_n(\mathbf{z}') = \sum_{k_1=0}^{\infty} \ldots \sum_{k_{q-1}=0}^{\infty} g(\mathbf{k}'; n) z_1^{k_1} \ldots z_{q-1}^{k_{q-1}}$, where $\mathbf{v}' = (v_1, \ldots, v_{q-1})$ is the truncation of a vector $\mathbf{v} = (v_1, \ldots, v_q)$ and ; is a right-cons operator defined by $\mathbf{v}'; x = (v_1, \ldots, v_{q-1}, x)$. $G_n(\mathbf{z}')$ relates to the generating functions of a network with one less class and $g(\mathbf{k})$ is the coefficient of $z_1^{k_1} \ldots z_{q-1}^{k_{q-1}}$ in $G_{k_q}(\mathbf{z}')$. We can repeat this argument recursively to obtain $g(\mathbf{k})$ in terms of a collection of single class normalising constant generating functions. It therefore remains to find an expression for $G_n(\mathbf{z}')$, which is given by the following proposition. We use the notation

$$\sum_{\sum_{l \neq i} j_l = j-1} \cdots$$

to denote a sum over a domain of vectors of length M in which the ith component is held fixed at an arbitrary value and the others are non-negative integers that sum to $j-1$. Of course, this is equivalent to a sum over a set of vectors of length $M-1$ without the constraint on the ith component, but our definition makes the indexing of terms in the summand much simpler.

Proposition 1 *In a closed q-class queueing network with M distinct node types, type i having multiplicity m_i, $1 \leq i \leq M$, and population of class j being k_j, $1 \leq j \leq q$, the generating function of the network's normalising constant when the class q population is fixed at n is:*

$$G_n(\mathbf{z}') = \sum_{i=1}^{M} \sum_{j=1}^{m_i} A_{ij}(\mathbf{z}') (-1)^{j+1} \frac{(m_i+n-j)!}{(m_i-j)!n!} \frac{\rho_{iq}^n}{(1-\boldsymbol{\rho}_i' \mathbf{z}')^{n+1+m_i-j}}$$

where

$$A_{ij}(\mathbf{z}') = \prod_{l=1, l \neq i}^{M} \left(\frac{\rho_{iq}}{(\rho_{iq}-\rho_{lq})(1-\frac{\rho_{iq}\boldsymbol{\rho}_l' - \rho_{lq}\boldsymbol{\rho}_i'}{\rho_{iq}-\rho_{lq}} \mathbf{z}')} \right)^{m_i} \sum_{\sum_{l \neq i} j_l = j-1} \prod_{l \neq i} s_l(j_l, \mathbf{z}')$$

and

$$s_l(k, \mathbf{z}') = \frac{(m_l + k - 1)!}{k!(m_l - 1)!} \left(\frac{\rho_{lq}}{(\rho_{iq} - \rho_{lq})(1 - \frac{\rho_{iq}\boldsymbol{\rho}'_l - \rho_{lq}\boldsymbol{\rho}'_i}{\rho_{iq} - \rho_{lq}}\mathbf{z}')} \right)^k$$

Proof
Let $G(\mathbf{z})$ be the generating function of the normalising constant of the q-class queueing network. By Cauchy's integral formula [Ahlfors 1996], p119, for example:

$$\begin{aligned} G_n(\mathbf{z}') &= \frac{1}{2\pi\sqrt{-1}} \oint_{C_q} z_q^{-n-1} G(\mathbf{z}) \mathrm{d}z_q \\ &= \frac{1}{2\pi\sqrt{-1}} \oint_{C_q} \frac{z_q^{-n-1}}{\prod_{i=1}^{M}(1 - \boldsymbol{\rho}_i \mathbf{z})^{m_i}} \mathrm{d}z_q \end{aligned} \tag{2}$$

where C_q is a closed contour around the origin in complex z_q-space which surrounds all the poles of $G(\mathbf{z})$. The integrand of equation 2 is analytic except at $z_q = \frac{1}{\rho_{iq}}(1 - \boldsymbol{\rho}'_i \mathbf{z}')$, for $i = 1, \ldots, M$, which we denote θ_{iq}. Cauchy's Residue Theorem [Ahlfors 1996] then yields:

$$G_n(\mathbf{z}') = -\sum_{i=1}^{M} r_i$$

where r_i is the residue at the point $z_q = \theta_{iq}$, which is given by:

$$r_i = \frac{1}{(m_i - 1)!} \frac{1}{(-\rho_{iq})^{m_i}} \left[\left(\frac{\partial}{\partial z_q} \right)^{m_i - 1} \frac{z_q^{-n-1}}{\prod_{j \neq i}(1 - \boldsymbol{\rho}_j \mathbf{z})^{m_j}} \right]_{z_q = \theta_{iq}}$$

By Leibnitz's rule for differentiating products, this equation can be re-written as:

$$r_i = \frac{1}{(-\rho_{iq})^{m_i}} \sum_{j=1}^{m_i} \frac{1}{(j-1)!(m_i - j)!} \left[\left(\frac{\partial}{\partial z_q} \right)^{j-1} \frac{1}{\prod_{j \neq i}(1 - \boldsymbol{\rho}_j \mathbf{z})^{m_j}} \right] \left(\frac{\partial}{\partial z_q} \right)^{m_i - j} z_q^{-n-1} \tag{3}$$

where, for simplicity, we do not indicate explicitly that the derivatives are evaluated at $z_q = \theta_{iq}$. Noting that

$$\left(\frac{\partial}{\partial z_q} \right)^l z_q^{-n-1} = (-1)^l \frac{(n+l)!}{n!} \frac{\rho_{iq}^{n+l+1}}{(1 - \boldsymbol{\rho}'_i \mathbf{z}')^{n+l+1}} \tag{4}$$

and substituting equation 4 into equation 3 and re-arranging, we obtain:

$$r_i = \sum_{j=1}^{m_i} \frac{(-1)^j \rho_{iq}^{-j+1}}{(j-1)!} \left[\left(\frac{\partial}{\partial z_q} \right)^{j-1} \frac{1}{\prod_{l \neq i}(1 - \boldsymbol{\rho}_l \mathbf{z})^{m_l}} \right] \frac{(n + m_i - j)!}{n!(m_i - j)!} \frac{\rho_{iq}^n}{(1 - \boldsymbol{\rho}'_i \mathbf{z}')^{n + m_i - j + 1}} \tag{5}$$

Similarly,

$$\left(\frac{\partial}{\partial z_q} \right)^l \frac{1}{(1 - \boldsymbol{\rho}_j \mathbf{z})^{m_j}} = \frac{(m_j + l - 1)!}{(m_j - 1)!} \frac{\rho_{jq}^l}{(1 - \boldsymbol{\rho}'_j \mathbf{z}' - \rho_{jq}\theta_{iq})^{m_j + l}}$$

and so, again using Leibnitz's rule, we obtain:

$$\left(\frac{\partial}{\partial z_q}\right)^{j-1} \frac{1}{\prod_{l\neq i}(1-\boldsymbol{\rho}_l\mathbf{z})^{m_l}} =$$

$$\sum_{\sum_{l\neq i} j_l = j-1} \frac{(j-1)!}{\prod_{l\neq i} j_l!} \prod_{l\neq i} \frac{(m_l+j_l-1)!}{(m_l-1)!} \frac{\rho_{lq}^{j_l}}{(1-\ \boldsymbol{\rho}_l'\mathbf{z}' - \rho_{lq}\theta_{iq})^{m_l+j_l}} \tag{6}$$

Substituting into equation 5, the result follows with a little algebra. □

Proposition 1 states that $g(\mathbf{k})$ is the coefficient of $z_1^{k_1}\ldots z_{q-1}^{k_{q-1}}$, which we abbreviate to $\mathbf{z}'^{\mathbf{k}'}$, in the generating function

$$G_{k_q}(\mathbf{z}') = \sum_{i=1}^{M}\sum_{j=1}^{m_i}(-1)^{j+1}\frac{(m_i+k_q-j)!\rho_{iq}^{k_q}}{(m_i-j)!k_q!}\ \frac{1}{(1-\ \boldsymbol{\rho}_i'\mathbf{z}')^{k_q+1+m_i-j}}\prod_{l\neq i}\left(\frac{\rho_{iq}}{\rho_{iq}-\rho_{lq}}\right)^{m_i}\times$$

$$\sum_{\sum_{l\neq i} j_l=j-1}\left[\prod_{l\neq i}\frac{(m_l+j_l-1)!\rho_{lq}^{j_l}}{j_l!(m_l-1)!(\rho_{iq}-\rho_{lq})^{j_l}}\right]\left[\prod_{l\neq i}\frac{1}{(1-\mathbf{x}_{li}'\mathbf{z}')^{m_i+j_l}}\right]$$

where, for $l \neq i$,

$$x_{li}' \quad = \quad \frac{\rho_{iq}\ \boldsymbol{\rho}_l' - \rho_{lq}\ \boldsymbol{\rho}_i'}{\rho_{iq}-\rho_{lq}}$$

We are free to define x_{ii}' as we like and we choose

$$x_{ii}' = \ \boldsymbol{\rho}_i'$$

If $\rho_{i_0q} = \rho_{l_0q}$ for some $i_0 \neq l_0$, the proposition's expression for $G_n(\mathbf{z}')$ appears to contain terms (at $i = i_0$) with zero denominator. These originate from $(1-\ \boldsymbol{\rho}_{l_0}'\mathbf{z}' - \rho_{l_0q}\theta_{i_0q})$ in equation 6 when $1-\rho_{l_0q}\theta_{i_0q}$ has no constant term, but there is no singularity since they combine with the terms $(1-\mathbf{x}_{l_0i_0}'\mathbf{z}')^{m_{i_0}+j_{l_0}}$ to give $1/(\rho_{i_0q}(\boldsymbol{\rho}_{i_0}'-\boldsymbol{\rho}_{l_0}')\mathbf{z}')^{m_{i_0}+j_{l_0}}$. However, this degeneracy does make the recurrence given in Theorem 1 below more complex. Further discussion is given in the conclusion of this paper.

We note that the generating function G depends on the M-vectors $\mathbf{m} = (m_1,\ldots,m_M)$ and $\mathbf{R} = (\boldsymbol{\rho}_1,\ldots,\boldsymbol{\rho}_M)$ and define, more precisely,

$$G(\mathbf{z}) \equiv H_q(\mathbf{M},\mathbf{R};\mathbf{z}) = \prod_{i=1}^{M}\frac{1}{(1-\boldsymbol{\rho}_i\mathbf{z})^{m_i}} \tag{7}$$

and

$$G_n(\mathbf{z}') \equiv H_{q;n}(\mathbf{M},\mathbf{R};\mathbf{z}') = \frac{1}{n!}\left[\frac{\partial^n H_q(\mathbf{M},\mathbf{R};\mathbf{z})}{\partial z_q^n}\right]_{z_q=0}$$

The following recurrence is now immediate.

Theorem 1 *In the notation of Proposition 1, if class q has distinct loads over the node types, i.e. if $\rho_{iq} \neq \rho_{jq}$ for $i \neq j$, then $g(\mathbf{k})$ is the coefficient of $\mathbf{z'}^{\mathbf{k}'}$ in*

$$H_{q;k_q}(\mathbf{M}, \mathbf{R}; \mathbf{z}') = \sum_{i=1}^{M} \sum_{j=1}^{m_i} (-1)^{j+1} \frac{(m_i + k_q - j)! \rho_{iq}^{k_q}}{(m_i - j)! k_q!} \prod_{l \neq i} \left(\frac{\rho_{iq}}{\rho_{iq} - \rho_{lq}} \right)^{m_i} \times$$
$$\sum_{\sum_{l \neq i} j_l = j-1} \left[\prod_{l \neq i} \frac{(m_l + j_l - 1)! \rho_{lq}^{j_l}}{j_l! (m_l - 1)! (\rho_{iq} - \rho_{lq})^{j_l}} \right] H_{q-1}(\mathbf{M} + \mathbf{J}, \mathbf{X}_i; \mathbf{z}')$$

where $\mathbf{J} = (j_1, \ldots, j_M)$, $\mathbf{X}_i = (x_{li} | l \neq i)$, vector addition is componentwise (i.e. $(\mathbf{M} + \mathbf{J})_i = m_i + j_i$) and, in the summations over $\mathbf{j}$, we fix $j_i = k_q + 1 - j$.

Thus, the coefficient of $\mathbf{z}^{\mathbf{k}}$ in H_q, $g(\mathbf{k})$, is equal to the corresponding sum of the coefficients of $\mathbf{z'}^{\mathbf{k}'}$ in $H_{q-1}(\mathbf{M} + \mathbf{J}, \mathbf{X}_i; \mathbf{z}')$.

In the special case that all nodes have distinct load vectors, i.e. $X = M$ and $m_i = 1$ for all i $(1 \leq i \leq M)$, the theorem simplifies to

$$H_{q;k_q}(\mathbf{M}, \mathbf{R}; \mathbf{z}) = \sum_{i=1}^{M} \rho_{iq}^{k_q} \prod_{l \neq i} \left(\frac{\rho_{iq}}{\rho_{iq} - \rho_{lq}} \right) H_{q-1}(\mathbf{1} + k_q \mathbf{e}_i, \mathbf{X}_i; \mathbf{z}')$$

where the M-component vectors $\mathbf{1} = (1, \ldots, 1)$ and $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ is the unit vector in the ith dimension.

3 Two class example

To illustrate our method, consider a network with two classes and two nodes whose generating function is given by:

$$G(z_1, z_2) = \frac{1}{(1 - \rho_{11} z_1 - \rho_{12} z_2)(1 - \rho_{21} z_1 - \rho_{22} z_2)}$$

Inverting this equation by using Theorem 1, or Proposition 1 directly, we obtain:

$$g(k_1, k_2) = g_1(k_1) + g_2(k_1)$$

where $g_1(k)$ and $g_2(k)$ are the normalising constants of single class networks with generating functions respectively:

$$G_1(\mathbf{z}) = \frac{-\rho_{12}^{k_2+1}}{[\rho_{22} - \rho_{12} - (\rho_{11}\rho_{22} - \rho_{12}\rho_{21})z](1 - \rho_{11} z)^{k_2+1}}$$

and

$$G_2(\mathbf{z}) = \frac{\rho_{22}^{k_2+1}}{[\rho_{22} - \rho_{12} - (\rho_{11}\rho_{22} - \rho_{12}\rho_{21})z](1 - \rho_{21} z)^{k_2+1}}$$

The required normalising constant $g(k_1, k_2)$ is then the coefficient of z^{k_1} in $G_1(\mathbf{z}) + G_2(\mathbf{z})$.

Note that the reduced networks are single class and contain one pole of multiplicity $k_2 + 1$ where k_2 is the population of class 2 jobs. The two class network has been replaced by two single class *pseudo*-networks—with standard solution methods—but the price to pay is the node with multiplicity dependent on the population of the replaced class. Obviously, when k_2 is large, a further partial fraction expansion is not recommended. We compared the numerical evaluation of equation 8 against evaluation using the convolution algorithm (see for example [Bruell and Balbo 1980, Harrison and Patel 1993] and observed that there was a reduction of almost 80% in processing time. Both methods are equally accurate.

For $\rho_{12} \neq \rho_{22}$ (say $\rho_{12} < \rho_{22}$), we define $\alpha = \frac{\rho_{11}\rho_{22}-\rho_{12}\rho_{21}}{\rho_{22}-\rho_{12}}$ and write

$$\begin{aligned} G_1(\mathbf{z}) &= \frac{-\rho_{12}^{k_2+1}}{(\rho_{22}-\rho_{12})(1-\alpha z)(1-\rho_{11}z)^{k_2+1}} \\ G_2(\mathbf{z}) &= \frac{\rho_{22}^{k_2+1}}{(\rho_{22}-\rho_{12})(1-\alpha z)(1-\rho_{21}z)^{k_2+1}} \end{aligned}$$

Then the coefficient of z^{k_1} in the expansion of G_2 dominates the corresponding coefficient in the expansion of G_1. Hence we only need consider G_2 in an asymptotic analysis. Suppose now that $\rho_{21} < \rho_{11}$ so that the class 1 load is higher at node 1 whereas (by our previous assumption) the class 2 load is higher at node 2. This is the interesting case since when both classes have higher load at the same node then that node will always be the bottleneck as the total network population increases (with whatever ratio between classes). With our assumptions, we expect node 1 to be the bottleneck when the proportion of the class 1 population is sufficiently high and conversely for class 2 and node 2. In between these proportions there might be a region where the nodes bottleneck simulataneously. We define $\beta = k_1/(k_1+k_2)$ to be the proportion of class 1 tasks.

The coefficient of z^{k_1} in $G_2(z)$ is the convolution

$$\frac{\rho_{22}^{k_2+1}\rho_{21}^{k_1}}{\rho_{22}-\rho_{12}} \sum_{i=0}^{k_1} \binom{k_2+i}{i} \left(\frac{\alpha}{\rho_{21}}\right)^i$$

Writing x_i for the ith term in the series, we have

$$g(k_1,k_2) \approx \rho_{21}\left[g(k_1-1,k_2) + \frac{\rho_{22}^{k_2+1}\rho_{21}^{k_1}}{\rho_{22}-\rho_{12}} x_{k_1}\right]$$

Now, the ratio $r_i = x_i/x_{i-1}$ is equal to $\frac{(k_2+i)\alpha}{i\rho_{21}}$ which is decreasing in i. Hence we have two cases:

1. $r_{k_1} < 1$, i.e. $\beta > \alpha/\rho_{21}$. In this case, terms in the series eventually decrease so that the last term can be considered negligible. Hence we have

$$g(k_1,k_2) \approx \rho_{21} g(k_1-1,k_2)$$

   The utilisation of class 1 at node 2, U_{21} is then $\frac{g(k_1-1,k_2)}{g(k_1,k_2)}\rho_{21} = 1$

2. $r_{k_1} > 1$, i.e. $\beta < \alpha/\rho_{21}$. Here terms increase throughout the series and so the last terms dominate. In reverse order starting from the last term, these terms approximate a geometric progression with parameter $r_{k_1}^{-1}$. Hence

$$g(k_1,k_2) \approx \frac{\rho_{22}^{k_2+1}\rho_{21}^{k_1} x_{k_1}}{(\rho_{22}-\rho_{12})(1-r_{k_1}^{-1})}$$

   so that

$$U_{21} \approx \frac{\rho_{21}}{\rho_{21} x_{k_1}} = \beta\frac{\rho_{21}}{\alpha}$$

Hence, asymptotically, class 1 utilisation at node 2 increases linearly with the proportion of class 1 tasks, β, until it reaches unity at $\beta = \alpha/\rho_{21}$, after which it remains there. Appying a similar argument for node 1 yields a corresponding threshold and between these values of β the nodes saturate simulataneously as the population of the network goes to infinity. These results are consistent with the observations of [Balbo and Serazzi 1996, Balbo and Serazzi 1997].

4 Conclusion

We have described an approach for computing the normalising constant in multiple class, closed, product-form queueing networks. The method is more efficient than the straightforward convolution algorithm and can provide insight into the asymptotic behaviour of such networks.

The most immediate theoretical extension of this work is the handling of the degeneracy that occurs when $\rho_{iq} = \rho_{lq}$ for some $i \neq l$ in Proposition 1, giving factors $1/(\rho_{iq}(\boldsymbol{\rho}'_i - \boldsymbol{\rho}'_l)\mathbf{z}')^{m_i+j_l}$. These factors do not conform with the structure of the product (1) and so cannot be reduced recursively by Theorem 1.

In the simplest case, for a given i, suppose that $\rho_{l_0 q} = \rho_{iq}$ for only one l_0; without loss of generality let that $l_0 = M$. We wish to find the coefficient of $\mathbf{z}'^{\mathbf{k}}$, for any vector $\mathbf{k}$ of $q-1$ non-negative components, in functions, $f(\mathbf{z}')$ say, of the form

$$f(\mathbf{z}') = \frac{1}{(\rho_{iq}(\boldsymbol{\rho}'_i - \boldsymbol{\rho}'_M)\mathbf{z}')^{n_M}} \prod_{l=1}^{M-1} \frac{1}{(1 - \boldsymbol{\nu}_l \mathbf{z}')^{n_l}}$$

for various $\boldsymbol{\nu}_l$ and n_l (actually $\mathbf{x}'_{li}$ and $m_i + j_l$ respectively in Proposition 1). In general there will be several instances of such functions $f(\mathbf{z}')$ in Proposition 1.

One approach to solving this problem is to make the following linear change of variable:

$$\begin{aligned} u_j &= z_j \quad (1 \le j \le q-2) \\ u_{q-1} &= \rho_{iq}(\boldsymbol{\rho}'_i - \boldsymbol{\rho}'_M)\mathbf{z}' \end{aligned}$$

Now,

$$\begin{aligned} \frac{\partial}{\partial z_h} &= \frac{\partial}{\partial u_h} + \frac{\partial u_{q-1}}{\partial z_h}\frac{\partial}{\partial u_{q-1}} \\ &= \frac{\partial}{\partial u_h} + \rho_{iq}(\rho_{ih} - \rho_{Mh})\frac{\partial}{\partial u_{q-1}} \quad (1 \le h \le q-2) \\ \frac{\partial}{\partial z_{q-1}} &= \rho_{iq}(\rho_{i,q-1} - \rho_{M,q-1})\frac{\partial}{\partial u_{q-1}} \end{aligned}$$

Now let ∂_h abbreviate $\frac{\partial}{\partial z_h}$, ∂'_h abbreviate $\frac{\partial}{\partial u_h}$ and a_h denote $\rho_{iq}(\rho_{ih} - \rho_{Mh})$ $(1 \le h \le q-1)$. Then

$$\begin{aligned} \partial_h^n &= \sum_{r=0}^{n} \binom{n}{r} a_h^r \partial'^r_{q-1} \partial'^{n-r}_h \quad (1 \le h \le q-2) \\ \partial_{q-1}^n &= a_{q-1}^n \partial'^n_{q-1} \end{aligned}$$

Extending the notation to vectors by defining

$$\partial^{\mathbf{n}} = \partial_1^{n_1}, \ldots, \partial_{q-1}^{n_{q-1}}$$

we have

$$\partial^{\mathbf{n}} = a_{q-1}^{n_{q-1}} \partial_{q-1}^{\prime n_{q-1}} \prod_{h=1}^{q-2} \sum_{r_h=0}^{n_h} \binom{n_h}{r_h} a_h^{r_h} \partial_{q-1}^{\prime r_h} \partial_h^{\prime n_h - r_h}$$

Hence the coefficient of $\mathbf{z}^{\mathbf{k}}$ in $f(\mathbf{z})$ is a linear combination of the coefficients of $\mathbf{u}^{\mathbf{v}}$ in $f(\mathbf{u})$ (i.e. f considered as a function of $\mathbf{u}$) for various vectors $\mathbf{v}$. But the coefficient of $\mathbf{u}^{\mathbf{v}}$ in $f(\mathbf{u})$ is the coefficient of $\mathbf{u}^{(v_1,\ldots,v_{q-1}-n_M)}$ in

$$\prod_{l=1}^{M-1} \frac{1}{(1-\boldsymbol{\nu}_l \mathbf{z}')^{n_l}}$$

This can be obtained by returning to the recurrence of Theorem 1.

References

[Ahlfors 1996] Ahlfors, L.V. *Complex analysis.* McGraw-Hill, 1979.

[Balbo and Serazzi 1996] Balbo, G. and G. Serazzi. "Asymptotic analysis of multiclass closed queueing networks: Common bottleneck." *Performance Evaluation* 26, 51–72, 1996.

[Balbo and Serazzi 1997] Balbo, G. and G. Serazzi. "Asymptotic analysis of multiclass closed queueing networks: Multiple bottlenecks." *Performance Evaluation* 30, 115–152, 1997.

[Birman et al. 1992] Birman, A.; D Ferguson and Y. Kogan. "Asymptotic Solutions for a Model of Large Multiprogramming Systems with Multiple Workloads." In *Computer Performance Evaluation '92*, R. Pooley and J. Hillston, eds., Antony Rowe, Chippenham, UK, 225–237, 1992.

[Bruell and Balbo 1980] Bruell, S.C. and G. Balbo. *Computational Algorithms for Closed Queueing Networks.* North-Holland, 1980.

[Guillemin 1949] Guillemin, E.A. *The Mathematics of Circuit Analyis.* John Willey & Sons, 1949.

[Harrison and Patel 1993] Harrison, P.G. and N.M. Patel. *Performance Modelling of Communication Networks and Computer Architectures.* Addison-Wesley Publishing Company, 1993.

[Knessl and Tier 1990] Knessl, C. and C. Tier. "Asymptotic expansions for large closed queueing networks." *Journal of the ACM* 37, no. 1, 144–174, 1990.

[Mitra and McKenna 1986] Mitra, D. and J. McKenna. "Asymptotic Expansions for Closed Markovian Networks with State Dependent Service Rates." *Journal of the ACM* 33, no. 3, 568–592, 1986.