ONLINE SUPPLEMENT

Algorithms for Time-Varying Networks of Many-Server Fluid Queues

by Yunan Liu and Ward Whitt

A. Overview

A.1. The Main Paper

Following Liu and Whitt (2011a,2012a), in the main paper we have developed, implemented and tested three algorithms to compute the transient performance functions in fluid queue networks (FQNets). The first two algorithms apply to $G_t/M_t/s_t + GI_t)^m/M_t$ FQNets with m queues and proportional routing. The first algorithm Alg(FPE) in §3 iteratively solves a single fixed point equation (FPE) for the vector of total arrival rate (TAR) functions over the entire time interval $[0, T]$. Since the operator in the FPE is a contraction, the iteration for calculating the TAR converges geometrically fast. Given the TAR at each queue, we can apply the fluid algorithm for single queues (FASQ) with M_t service reviewed in §2 to each queue separately.

The second algorithm Alg(ODE) in §4 finds the TAR vector in any interval for which no queue changes its regime by solving an m-dimensional ordinary differential equation (ODE). However, just as for the FASQ, it is necessary to control the switching from one network regime to another. Both Alg(FPE) and Alg(ODE) have been shown to be effective. Alg(ODE) is appealing because it is easier to implement and is faster for small networks, but Alg(FPE) has been found to be more efficient for large networks, having run time of order $O(m)$ as compared to $O(m^2)$ for Alg(ODE).

The third algorithm Alg(FPE,GI) in §5 is a new algorithm to analyze FQNets with non-exponential service at each queue. This algorithm is less efficient computationally, but is appealing because non-exponential service times arise in many applications. Like algorithms FASQ and Alg(ODE), Alg(FPE,GI) requires working over intervals with a fixed network regime, so that the algorithm exploits a regime switching step size. For the individual queues with GI service, we apply the algorithm for the single fluid queue with GI service developed in Liu and Whitt (2012a), which is more complicated than the FASQ in §2 because it involves another FPE to calculate the flow rate into service. Fortunately, the operators in both these FPE's are contractions, so that iterations converge geometrically fast.

In §6 we described results of implementing and testing these algorithms for FQNets. The examples showed that all algorithms are effective and can provide useful approximations for corresponding stochastic queueing networks (SQNets) of many-server queues, experiencing periods of overloading.

A.2. The Contents of this Supplement

This is a supplement to the main paper, with material expanding on the paper presented in the order of the sections in the main paper. At the outset in §1, we assumed that our staffing functions are feasible, never forcing fluid out of service. We start in §B by discussing how to detect the first violation of feasibility in the FQNets, if any, and how to find the minimum feasible staffing function greater than or equal to an initial one if that one is infeasible. Next in §C we present additional material on the algorithm for single fluid queues in §2. We give a formal statement of the algorithm and we examine the running time as a function of the interval length, T, the number of switches, $\mathcal{S}$, and the switching step size, ΔT.

We give formal statements of the FPE-based and ODE-based algorithms for FQNets in §§D and E. Finally, we provide more results for the examples in §G. In §G.1.2 we give an example with a time-varying staffing function, illustrating that it can be effectively treated as well.

B. Detecting Staffing Function Feasibility

We now discuss how to detect the first violation of feasibility of a staffing function and how to find the minimum feasible staffing function greater than or equal to the original staffing function if that one is infeasible.

In general, there is no guarantee that a staffing function s is feasible, i.e., having the property that the staffing function is set exogenously and adhered to, without forcing any fluid that has entered service to leave without completing service, because we allow s to decrease. (The fluid is assumed to be incompressible.) At any time $t > 0$ and $i \in \mathcal{O}(t)$, we require

$$b_i(t,0) = s_i'(t) + \sigma_i(t) \geq 0. \tag{B.1}$$

We note that the above criterion becomes $s_i'(t) + \mu_i(t)\, s_i(t) \geq 0$ for all $i \in \mathcal{O}(t)$ for M_t service. Hence, we immediately have a sufficient condition for M_t service:

$$s_i'(t) + \mu_i\, s_i \geq 0, \quad \text{for all } 1 \leq i \leq m.$$

However, since we allow s_i to decrease, the feasibility condition in (B.1) might be violated in general when the given staffing functions decrease too quickly, i.e., when $-s_i' > 0$ is too large. The detection of infeasibility and construction of feasible staffing for a single fluid queue has already been discussed in §9 of Liu and Whitt (2012a). Also see Appendix G.2 there for an example. We next explain how to generalize our approach to the algorithms for the FQNets.

The idea is in the same spirit as in Liu and Whitt (2012a). Suppose in the kth iteration, the algorithm considers the time interval $\mathcal{I}_k \equiv [t^*, t^* + \Delta T]$. If there exists a time $t' \in \mathcal{I}_k$ such that the condition in (B.1) is violated for some $i \in \mathcal{O}(t')$, then we set $b_i(t, 0) = 0$ (shut the flow from the queue into the service facility) starting from time t'. Of course, the resulting staffing function $s_i^*(t) = B(t)$ will be strictly greater than $s_i(t)$. We continue with this strategy until our revised staffing function s_i^* coincides with the original s_i (if ever before T). See §9 and Appendix G.2 of Liu and Whitt (2012a) for more discussion.

For the FPE algorithm, we follow this strategy in each FPE iteration when we call FASQ with a temporary TAR $\lambda^{(k)}$, $k \geq 1$. For the ODE and FPE-GI algorithms, in each iteration interval I_j, we check the feasibility condition (B.1) for every $i \in \mathcal{O}(t)$, $i \in I_j$, and construct revised staffing functions if needed.

C. More on the Single-Queue Algorithm

We first give a formal statement of the single fluid-queue algorithm (FASQ) from Liu and Whitt (2011a,2012a) that was reviewed in §2.3. For each new regime switch time t, we use k to set up all computations needed from time t until the end of the interval at time T, but restart by selecting a new larger starting time whenever a regime switch is detected.

C.1. Sensitivity to the Switching Step Size

We now see how the FASQ run time depends on the switching step size ΔT. For that purpose, consider an $M_t/M/s_t + M$ queue over the time interval $[0, T]$ for $T = 20$ with a sinusoidal arrival rate $\lambda(t) = a + b \sin(c \cdot t)$, exponential service and abandonment distributions

Algorithm 1 : A Fluid Algorithm for Single Queues (FASQ) for the $G_t/M_t/s_t + GI_t$ fluid model, with model data $\mathcal{D} \equiv \left(\lambda, s, G, F, \hat{\mathcal{P}}(0)\right)$ and switching step size ΔT

```
1: Initialization: Set $\mathcal{R}(0)$ and let $t := 0$
2: **repeat**
3:   **for** $k = 0, 1, \ldots, \lceil (T - t)/\Delta T \rceil$ **do**
4:     Given $\mathcal{R}(t) = UL$, compute $(b, B)$ in interval $[t + (k-1)\Delta T, t + k\Delta T]$ using (2.18);
5:     Given $\mathcal{R}(t) = OL$, compute $\mathcal{P}$ in interval $[t+(k-1)\Delta T, t+k\Delta T]$ using (2.19)-(2.22), (2.2)-(2.4);
6:     **if** $T_{\mathcal{R}}(t) < t + k\Delta T$, **then**
7:       $t := T_{\mathcal{R}}(t)$
8:       $\mathcal{R} := \{\text{OL}, \text{UL}\} \backslash \mathcal{R}$
9:       BREAK for-loop
10:    **end if**
11:  **end for**
12: **until** $t \geq T$.
```

$G(x) = 1 - e^{-\mu x}$ and $F(x) = 1 - e^{-\theta x}$, constant staffing $s(t) = s$, with $a = c = \mu = s = 1$, $b = 0.6$ and $\theta = 0.5$. We manually made the system switches for about 100 times between UL and OL intervals (i.e., $\mathcal{S} = O(100)$) by setting $c = 30$ so that the arrival-rate period is $\tau = 2\pi/c = 0.21$. The total number of periods in $[0, 20]$ is $N(\tau) = 20/\tau_1 = 95.2$.

In Table 4, we provide the computation times $\mathcal{T}(\Delta T)$ as a function of ΔT for $0.02 \leq \Delta T \leq T = 20$.

ΔT	20	10	6.67	5	4	2	1.33	1
$\mathcal{T}(\Delta T)$	15.36	12.89	10.25	7.99	6.62	3.57	3.54	1.93
ΔT	0.67	0.5	0.33	0.25	0.2	0.1	0.04	0.02
$\mathcal{T}(\Delta T)$	1.93	1.93	1.93	1.93	1.93	1.93	1.94	1.94

Table 4: The computation time $\mathcal{T}(\Delta T)$ (seconds) as a function of ΔT in the interval $[0, T]$, $T = 20$.

We observe that $\mathcal{T}(\Delta T)$ is almost insensitive to the choices of ΔT as long as ΔT is not too big. First, clearly it is not efficient to have a big ΔT when $\mathcal{S}$ is large. Suppose the current interval is $[t, t + \Delta T]$. If the system changes regime right after t, say at $t + h$, then all computation for the performance functions in $[t + h, t + T]$ is wasted and has to be redone later (possibly many times). Second, if ΔT is small, the computation obviously becomes efficient for large $\mathcal{S}$. Third, if we choose a small ΔT when $\mathcal{S}$ is small, for instance, the system stays OL (or UL) for the entirely interval $[0, T]$, the algorithm could experience a large number of iterations $N(\Delta T) \equiv T/\Delta T$ before reaching time T. However, since the computation complexity in each iteration is linear in ΔT, the total complexity will be $(T/\Delta) \cdot O(\Delta T) = O(T)$, which is independent with ΔT. Hence, consistent with Table 4, we conclude that it should always be good to have a small ΔT regardless of the number of regime changes $\mathcal{S}$.

C.2. Sensitivity to $\mathcal{S}$ and T

We now see how the FASQ run time depends on the number of switches, $\mathcal{S}$, and the length of the time interval T. We consider the dependence upon $\mathcal{S}$ for fixed T and the dependence upon T for fixed $\mathcal{S}$. To do so, we use the same example considered in §C.1.

Table 5 gives the computation times $\mathcal{T}(\mathcal{S})$ for $1 \leq \mathcal{S} \leq 96$ in the interval $[0, T]$ with $T = 20$ by varying c. Here $\Delta T = 0.5$.

$\mathcal{S}$	1	2	4	7	13	20
$\mathcal{T}(\mathcal{S})$	0.98	0.75	0.76	1.13	1.76	2.50
$\mathcal{S}$	26	32	48	64	80	96
$\mathcal{T}(\mathcal{S})$	3.25	4.09	6.32	8.91	11.51	15.32

Table 5: The computation time $\mathcal{T}(\mathcal{S})$ (seconds) as a function of $\mathcal{S}$, the number of regimes switches (between UL and OL) using FASQ in the interval $[0, T]$, $T = 20$.

Next, Tables 6 and 7 give the computation times $\mathcal{T}(T)$ for $1 \leq T \leq 300$ in two cases: (i) with $c = 10\pi/T$ and (ii) with $c = 1$. We choose $\Delta T = T/150$. In case (i), we fixed the number of switches $\mathcal{S} \approx 5$ by decreasing c; in case (ii), the number of switches $\mathcal{S}$ grows linearly with T since c is a constant.

$\mathcal{S}$	5	10	15	20	25	30	40	50
$\mathcal{T}(T)$	0.267	0.32	0.71	0.97	1.25	1.53	2.12	2.72
$\mathcal{S}$	60	80	100	120	150	200	250	300
$\mathcal{T}(T)$	3.33	4.56	5.78	6.99	8.82	11.85	14.89	17.85

Table 6: The FASQ computation time $\mathcal{T}(T)$ (seconds) as a function of the length of time interval, T, for fixed total number of switches.

We plot $\mathcal{T}(\mathcal{S})$ as a function of $\mathcal{S}$ for fixed T in Figure 4(a) and $\mathcal{T}(T)$ as a function of T for fixed $\mathcal{S}$ in Figure 4(b) for case (i) and in Figure 5 for case (ii). Figure 4 shows

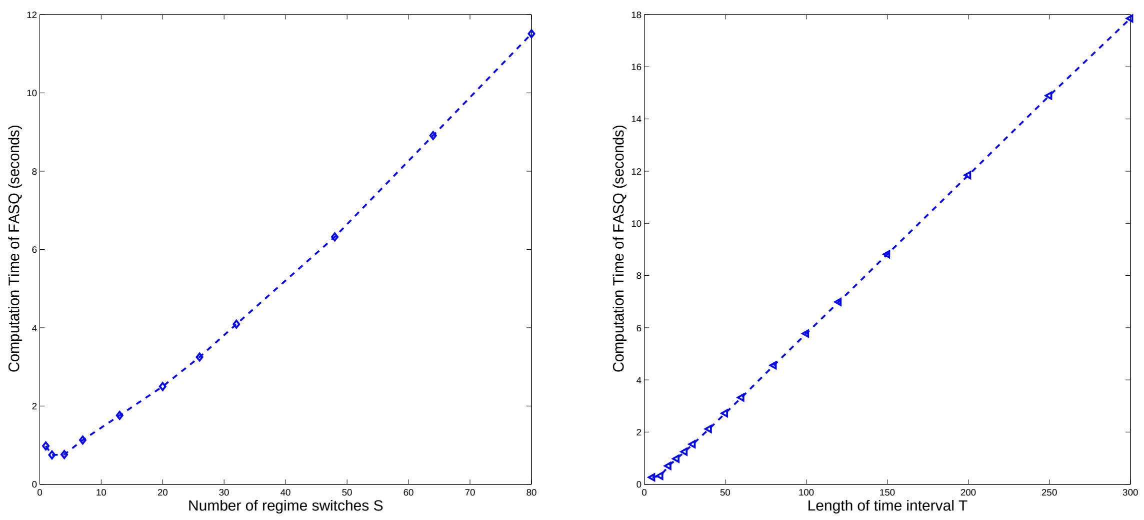

Figure 4: Computation times of FASQ in case (i) where $\mathcal{S}$ is independent of T, as functions of (a) the number of switches $\mathcal{S}$ for fixed T and (b) the length of time interval T for fixed $\mathcal{S}$.

that the computation time is linear in $\mathcal{S}$ (when T is fixed) and linear in T (when $\mathcal{S}$ is fixed). Figure 5 shows that the computation time is quadratic in T when $\mathcal{S} = O(T)$, because $\mathcal{T}(\mathcal{S}\,T) = O(\mathcal{S}\,T) = O(T^2)$. These experiment support our observations in §2.3.

$\mathcal{S}$	5	10	15	20	25	30	40	50
$\mathcal{T}(T)$	0.26	0.30	0.48	0.76	1.10	1.50	2.42	3.61
$\mathcal{S}$	60	80	100	120	150	200	250	300
$\mathcal{T}(T)$	4.98	8.40	12.71	17.94	27.34	47.54	73.33	104.31

Table 7: The computation time $\mathcal{T}(T)$ (seconds) as a function of the length of time interval, T, when $\mathcal{S}$, the number of switches, is proportional to T.

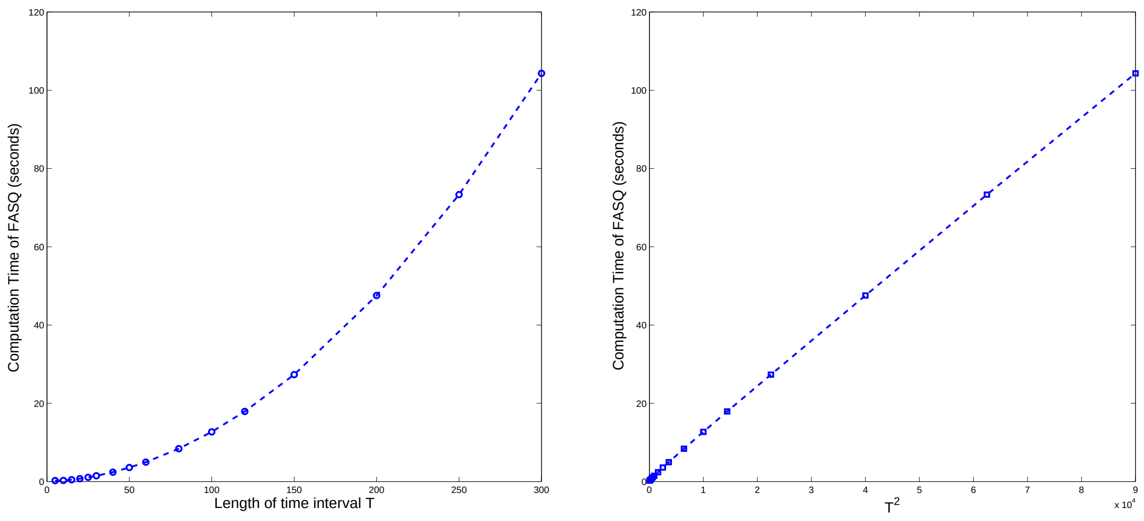

Figure 5: Computation times of FASQ in case (ii) where $\mathcal{S}$ is proportional to T, as functions of (a) T, the length of time interval, and (b) T^2.

D. The FPE-Based Algorithm

We now give a formal statement of the FPE-based algorithm in §3.

Algorithm 2 : An FPE based algorithm for the $(G_t/M_t/s_t + GI_t)^m/M_t$ Fluid Network

```
1: Initialization: $\lambda^{(1)} := \lambda^{(0)}$, $0 \le i \le m$,
2: for $k = 1, 2, \ldots$ do
3:     for $i = 1, 2, \ldots, m$ do
4:         Compute $\sigma_i$ in $[0,T]$ using FASQ (Algorithm 1) with data $\left(\lambda_i^{(k)}, s_i, G_i, F_i, \hat{\mathcal{P}}_i(0)\right)$
           and switching step size $\Delta T$
5:     end for
6:     Let $\lambda^{(k+1)} := \lambda^{(0)} + P^T \cdot \sigma$ in $[0,T]$
7:     if $\|\lambda^{(k+1)} - \lambda^{(k)}\|_T < \epsilon$ then
8:         $\lambda := \lambda^{(k+1)}$
9:         Break
10:    end if
11: end for
12: Compute $\mathcal{P}_i$ for $1 \le i \le m$ using FASQ (Algorithm 1) with data $\left(\lambda_i, s_i, G_i, F_i, \hat{\mathcal{P}}_i(0)\right)$
    and switching step size$\Delta T$.
```

E. The ODE-Based Algorithm

The ODE-based algorithm in §4 obtains the TAR vector over any interval over which there are no regime switches at any queue by solving an ODE. Thus, a key step is identifying successive intervals during which all queues remain in the same regime. Paralleling the FASQ, that is done with a network switching step size ΔT.

We now summarize the algorithm. It requires that we specify the desired time interval $[0,T]$, the vector of model data defined in (3.4), and a positive network switching step size ΔT (which should typically be shorter than for a single queue).

F. The FPE Algorithm for *GI* Service

We now present more about the algorithm for GI service distributions. Given a desired duration T of an interval $[0,T]$, the vector of the model data defined as (3.4), a step size $0 < \Delta T \le T$, and an error tolerance parameter (ETP) $\epsilon > 0$, we summarize the algorithm formally as the following.

Algorithm 3 : An ODE based algorithm for the $(G_t/M_t/s_t + GI_t)^m/M_t$ Fluid Network

```
1: Initialization: $t := 0$
2: **repeat**
3:    **for** $k = 0, 1, \ldots, \lceil (T - t)/\Delta T \rceil$ **do**
4:       Compute $\lambda(s)$ and $\mathbf{B}(s)$ for $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$, using (4.3)-(4.4)
5:       Compute $\mathcal{P}(s)$ for $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$ using (2.18)-(2.22), (2.2)-(2.4)
6:       **if** $T_{\mathcal{R}}(t) < t + k\,\Delta T$ for $T_{\mathcal{R}}(t)$ in (4.6) **then**
7:          $t := T_{\mathcal{R}}(t)$
8:          Update $\mathcal{U}(t)$ and $\mathcal{O}(t)$ by (4.1)-(4.2)
9:          BREAK for-loop
10:      **end if**
11:   **end for**
12: **until** $t \geq T$
```

Algorithm 4 : An FPE based algorithm for the $(G_t/GI/s_t + GI_t)^m/M_t$ Fluid Network

```
1: Initialization: $t := 0$
2: **repeat**
3:    **for** $k = 0, 1, \ldots, \lceil (T - t)/\Delta T \rceil$ **do**
4:       **for all** $i \in \mathcal{O}(t)$ **do**
5:          - Compute $b_i(s, 0)$ solving FPE (5.3) with ETP $\epsilon$, $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$
6:          - Let $\sigma_i(s) := b_i(s, 0) - s_i'(s)$
7:       **end for**
8:       Compute $\lambda(s)$ using FPE (5.5) with ETP $\epsilon$, $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$
9:       Compute $\mathcal{P}(s)$ for $s \in [t + (k-1)\Delta T, t + k\,\Delta T]$ using (2.18)-(2.22), (2.2)-(2.4) or
         the algorithm from Liu and Whitt (2012a) if the service is GI
10:      **if** $T_{\mathcal{R}}(t) < t + k\,\Delta T$ for $T_{\mathcal{R}}(t)$ in (4.6) **then**
11:         $t := T_{\mathcal{R}}$
12:         Update $\mathcal{U}(t)$ and $\mathcal{O}(t)$ by (4.1)-(4.2)
13:         BREAK for-loop
14:      **end if**
15:   **end for**
16: **until** $t \geq T$
```

G. More for the Examples

G.1. More on the Two-Queue FQNet

We demonstrate how the FPE-based algorithm works. Since it is key to obtain the total arrival rates $\lambda_1(t)$ and $\lambda_2(t)$ for $0 \leq t \leq T$, we first demonstrate how fast the fixed-point algorithm converges. We initially let $\lambda_i^{(1)}$ be $\lambda_i^{(0)}$, $i = 1, 2$. In Figure 6, we plot the total arrival rates in every iteration. The two functions at the bottom are $\lambda_1^{(0)}(t)$ and $\lambda_2^{(0)}(t)$; the functions at the top are the $\lambda_1(t)$ and $\lambda_2(t)$, computed using the ODE based algorithm; the other functions are the intermediate values computed using the FPE based algorithm. The monotone convergence and geometric rate of convergence are evident from Figure 6.

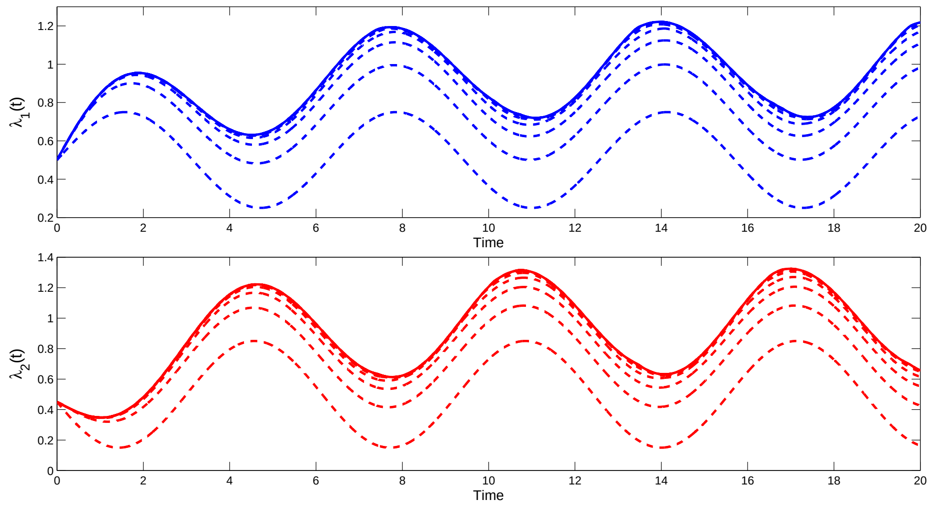

Figure 6: The convergence to the fixed point of the total arrival rate vector: the increasing computed values at each queue in successive iterations.

In Figure 7, we plot all standard performance measures of the fluid network using the FPE based algorithm, including λ_i, Q_i, w_i, B_i, X_i, and $b_i(\cdot, 0)$, $i = 1, 2$.

Figure 8 illustrate how these approximations perform for the same example with $n = 50$. Now the solid lines are simulation estimates of the mean of these scaled stochastic processes, obtained by averaging multiple independent sample paths. For this small value of n, the stochastic variability cannot be simply ignored. But the means can be well approximated by the fluid functions. Figure 1 and 8 show that the fluid approximation is effective in describing the performance of the stochastic system.

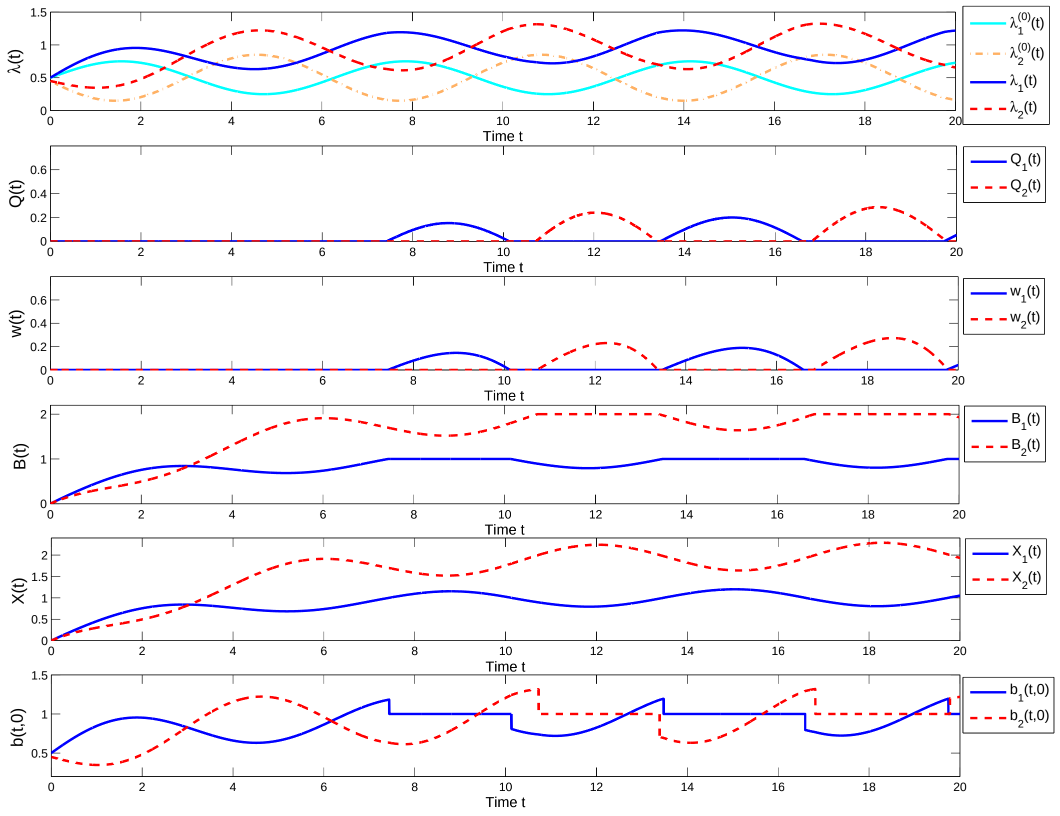

Figure 7: Computing the fluid performance functions for the $(M_t/M/s_t + M)^2/M_t$ network fluid model.

G.1.1. Different Phases

Complementing the two-queue example in §6.2, we now consider the same model with different phases in the sinusoidal arrival-rate functions. In particular, let $\phi_1 = 0$ and $\phi_2 = 1$ and let all other model parameters remain the same. As the analogs of Figure 7 and 8 (with $\phi_2 = -3$), we plot the fluid function and perform simulation comparison in Figure 9 and 10 with $\phi_2 = 1$. In this case the two queues become OL and UL almost at the same time.

G.1.2. Time-Varying Staffing

The two-queue example in §6.2 had constant staffing functions. To show that the algorithms can also handle the important feature of time-varying staffing, we now consider sinusoidal staffing, using the staffing functions

$$s_i(t) = \alpha_i + \beta_i \cdot \sin(\gamma_i\, t + \psi_i), \quad i = 1, 2, \tag{G.1}$$

with $\alpha_1 = 1$, $\alpha_2 = 2$, $\beta_1 = 0.6\,\alpha_1$, $\beta_2 = 0.5\,\alpha_2$, $\gamma_1 = 1$, $\gamma_2 = 0.5$, $\psi_1 = 3$, $\psi_2 = 0$. While other model parameters remain the same. With these particular choices on the model parameters,

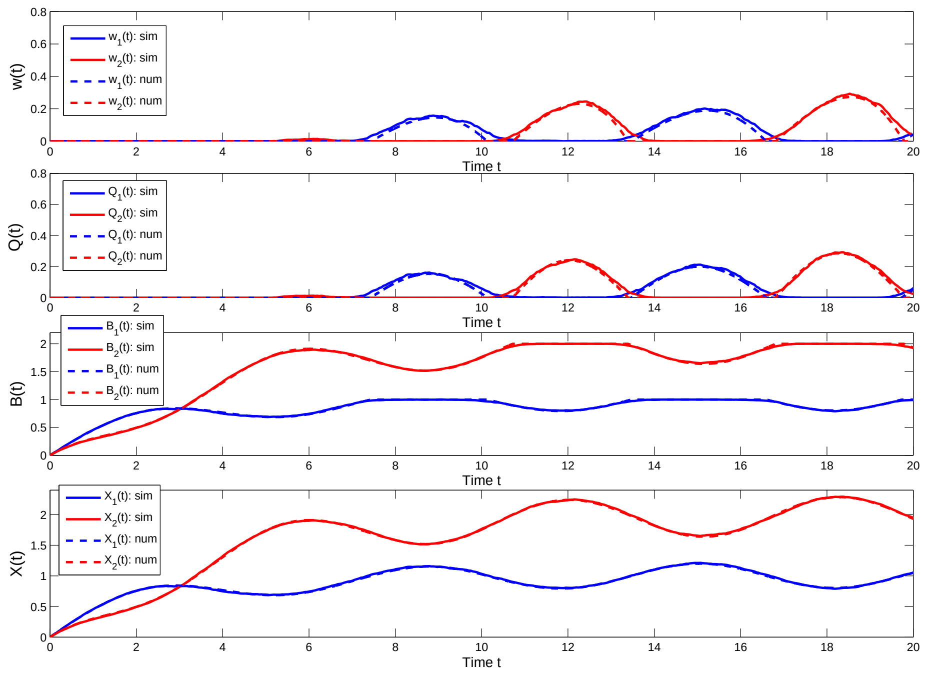

Figure 8: A comparison of performance functions in the $(M_t/M/s_t + M)^2/M_t$ FQNet with simulation estimates of time-varying mean values, obtained by averaging $n = 50$ independent sample paths from the corresponding QNet.

it is not hard to see that for $i = 1, 2$,

$$s_i'(t) + \mu_i\, s_i(t) = \beta_i\gamma_i \cos(\gamma_i\, t + \psi_i) + \mu_i\,(\alpha_i + \beta_i \sin(\gamma_i\, t + \psi_i)) \geq 0,$$

or equivalently,

$$\sin\left(\gamma_i\, t + \psi_i + \arctan\left(\frac{\gamma_i}{\mu_i}\right)\right) \geq -\frac{\mu_i\alpha_i}{\sqrt{\mu_i^2 + \gamma_i^2}}, \quad \text{for all} \quad t \geq 0,$$

which guarantees the feasibility for both staffing functions s_1 and s_2. See the Appendix I.2.1 of Liu and Whitt (2012a) for more discussion.

As the analogs of Figure 9, we plot the fluid function in Figure 11.

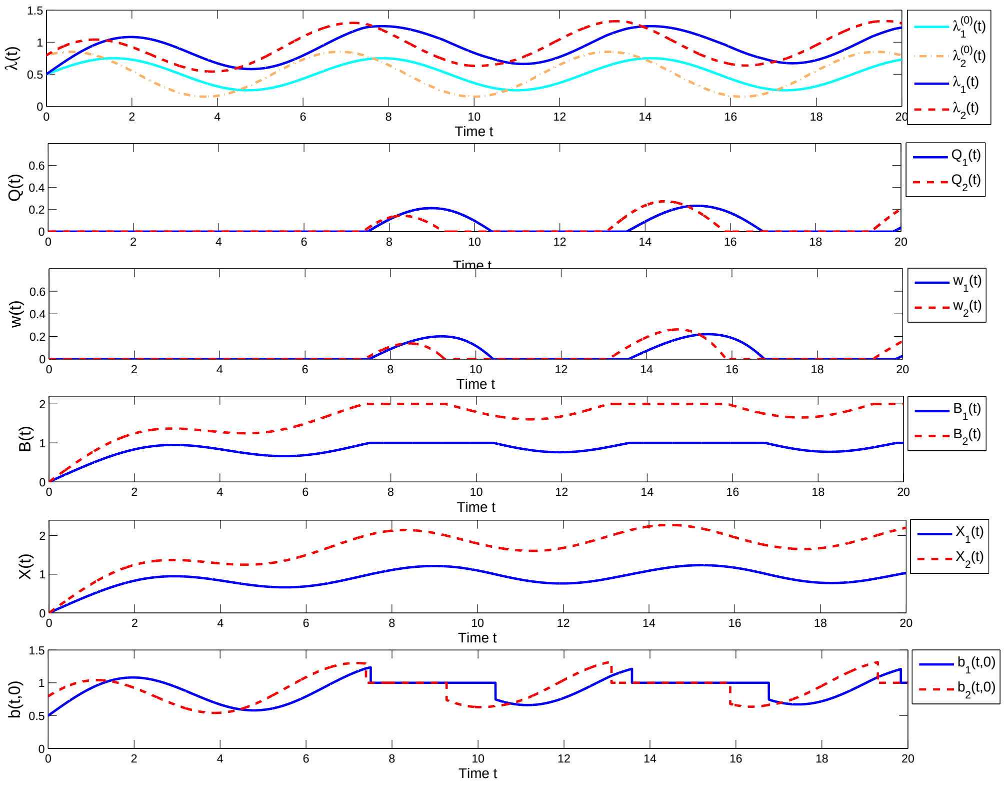

Figure 9: Computing the fluid performance functions for the $(M_t/M/s_t + M)^2/M_t$ network fluid model.

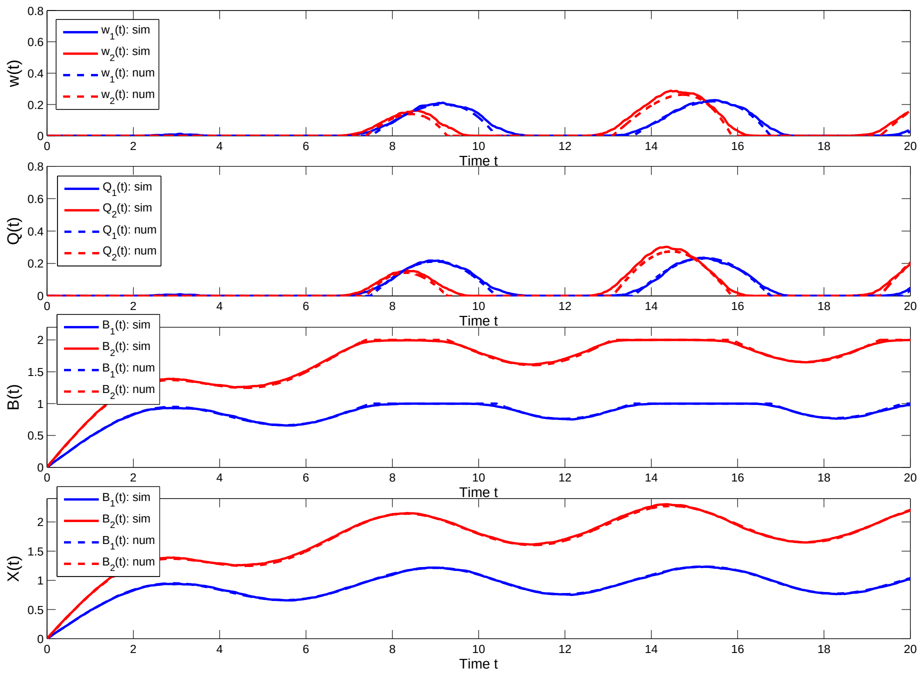

Figure 10: A comparison of the $(M_t/M/s_t + M)^2/M_t$ network fluid model with a simulation run averaging 50 independent sample paths, $n = 100$.

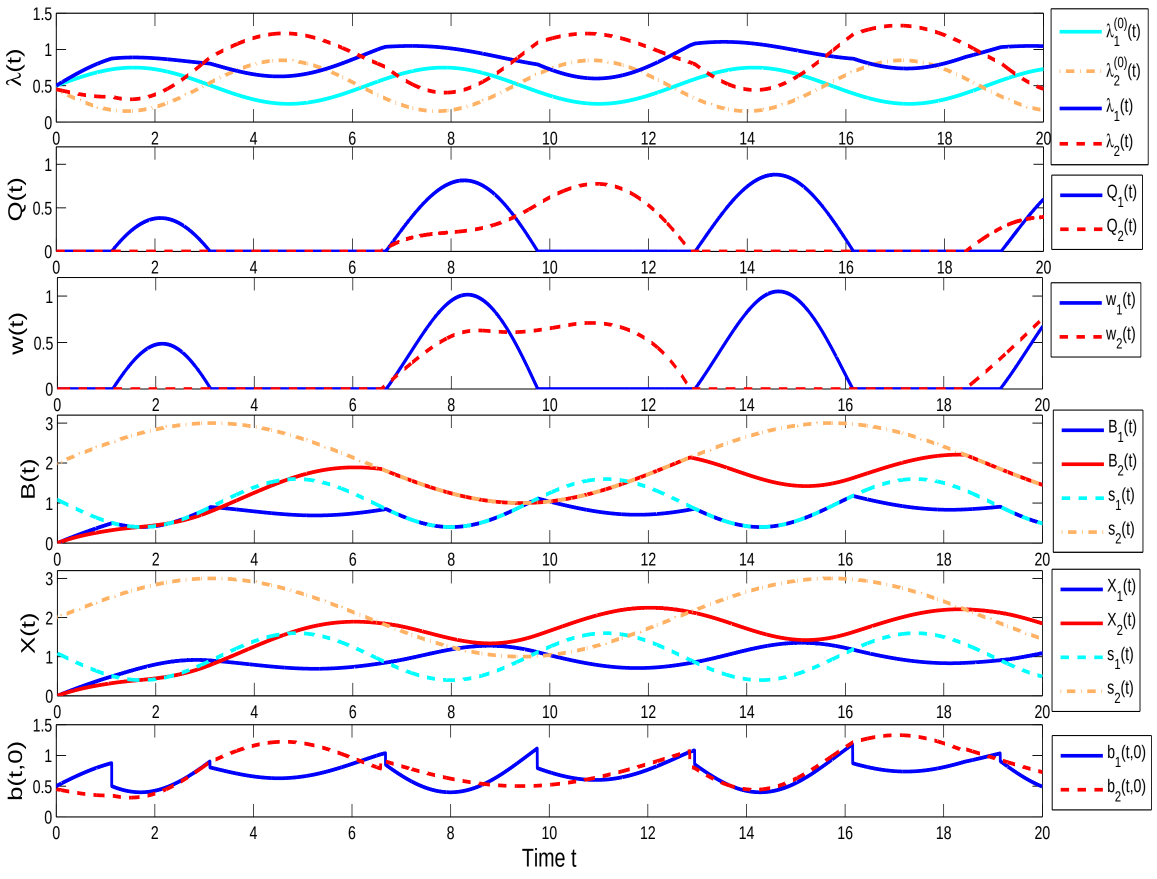

Figure 11: Computing the fluid performance functions for the $(M_t/M/s_t+M)^2/M_t$ network fluid model with sinusoidal staffing functions.

G.2. More on the Many-Queue Example

We now provide more material related to the many-queue example in §6.3.

Complementing the example in §6.3, we let $m = 10$, $\phi_i = 0$, $i = 1, \ldots, 10$. We repeat the experiment and plot the fluid functions for all 10 queues in Figure 12 (which is an analog of Figure 13). Figure 13 shows plots of the performance functions for $m = 10$. The red dashed lines are λ_i, i.e., the total arrival rates which are the solutions to the FPE; the blue solid lines are $\lambda_i^{(0)}$, i.e., the external arrival rates.

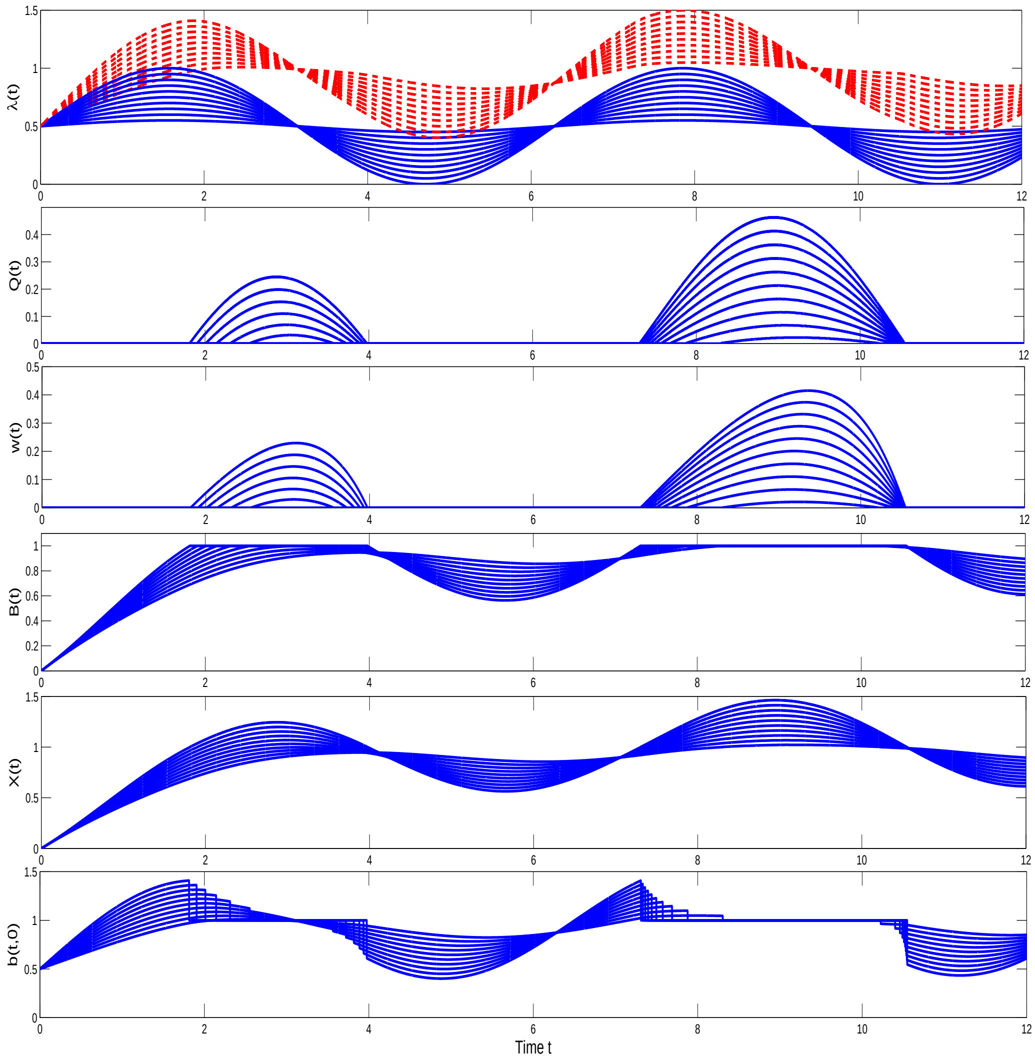

Figure 12: Computing the fluid performance functions for the $(M_t/M/s_t+M)^{10}/M_t$ network fluid model.

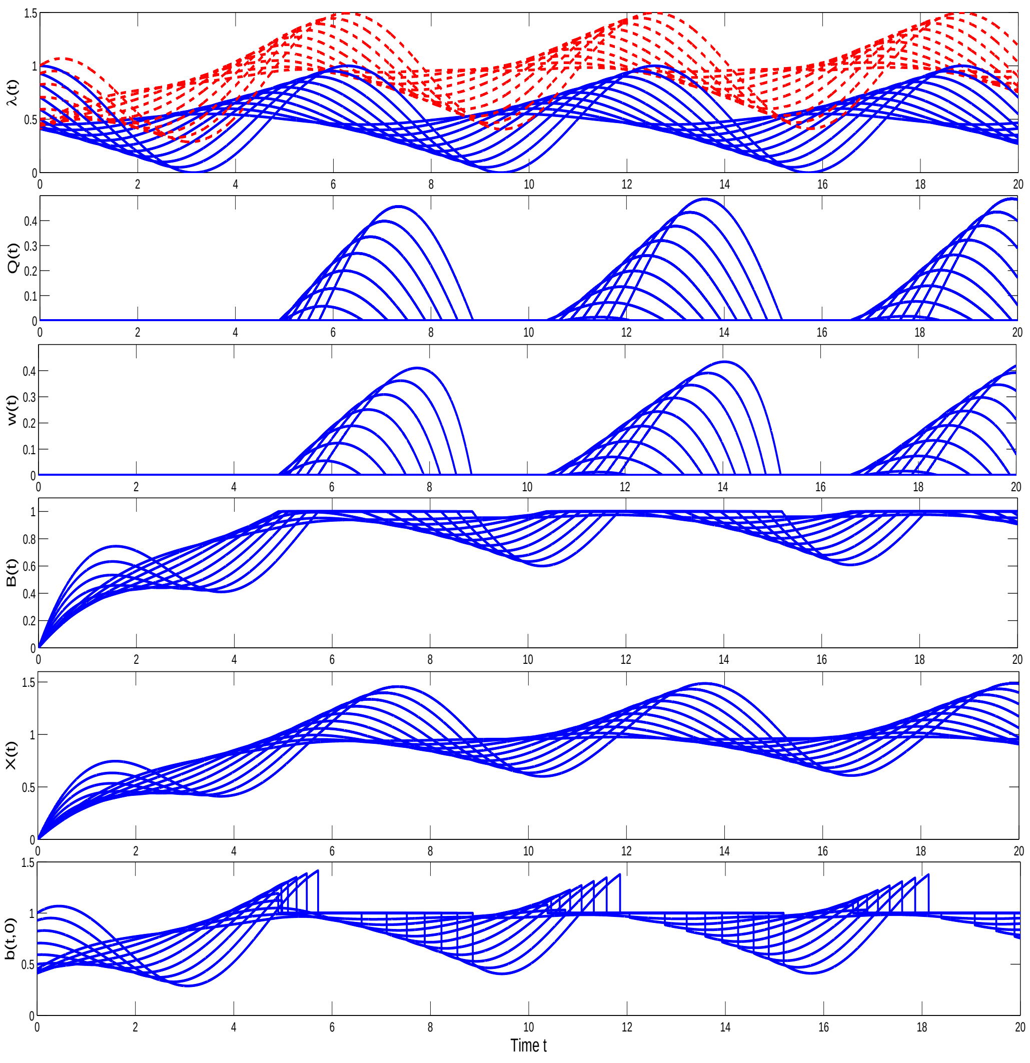

Figure 13: Computing the fluid performance functions for the $(M_t/M/s_t+M)^{10}/M_t$ network fluid model ($\phi_i = \pi(1.5 - i/m)$).

G.3. More on the *GI* Service Example

Figure 14 compares the results of Alg(FPE,GI) applied to the $(M_t/LN/s+E_2)^2/M$ FQNet with simulation estimates of mean values for the corresponding stochastic processes in the corresponding SQNet with $n = 50$, obtained by averaging the sample paths from 200 independent replications.

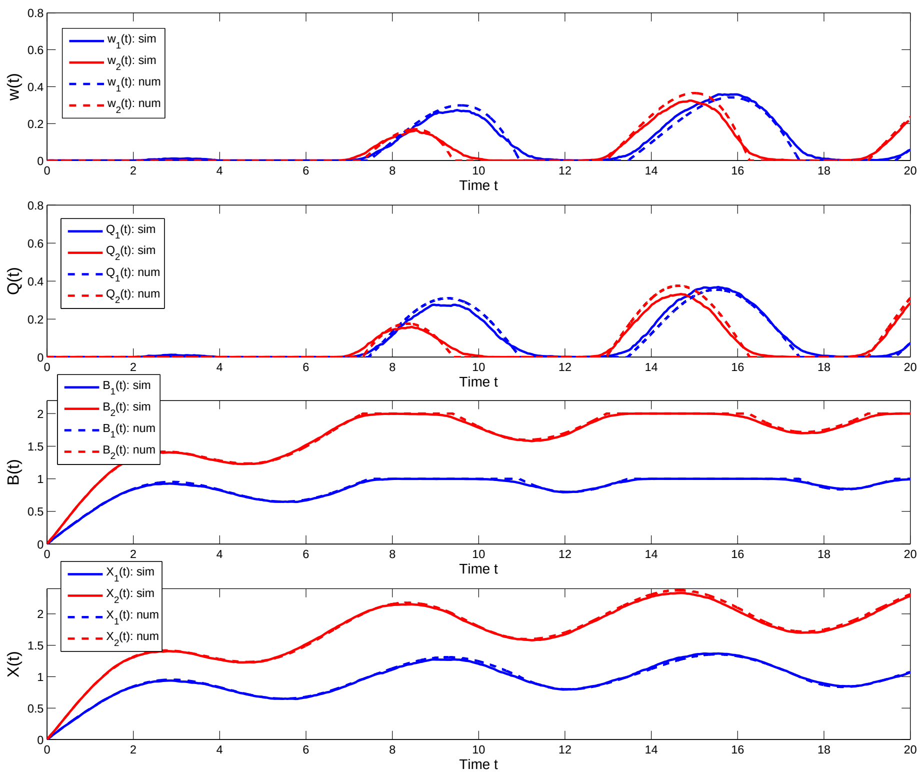

Figure 14: A comparison of the $(M_t/LN/s_t + E_2)^2/M_t$ network fluid model with a simulation run averaging 50 independent sample paths, $n = 100$.