

Original citation:

MacKay, Robert S. and Robinson, J. D. (2018) *Aggregation of Markov flows I : theory*. *Philosophical Transactions A: Mathematical, Physical and Engineering Sciences*, 376 (2118). doi:[10.1098/rsta.2017.0232](https://doi.org/10.1098/rsta.2017.0232)

Permanent WRAP URL:

<http://wrap.warwick.ac.uk/97255>

Copyright and reuse:

The Warwick Research Archive Portal (WRAP) makes this work of researchers of the University of Warwick available open access under the following conditions. Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher statement:

First published by Royal Society of Chemistry 2018
<http://dx.doi.org/10.1098/rsta.2017.0232>

A note on versions:

The version presented here may differ from the published version or, version of record, if you wish to cite this item you are advised to consult the publisher's version. Please see the 'permanent WRAP URL' above for details on accessing the published version and note that access may require a subscription.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk

AGGREGATION OF MARKOV FLOWS I: THEORY

R.S.MACKAY AND J.D.ROBINSON

ABSTRACT. A Markov flow is a stationary measure, with the associated flows and mean first passage times, for a continuous-time regular jump homogeneous semi-Markov process on a discrete state-space. Nodes in the state-space can be eliminated to produce a smaller Markov flow which is a factor of the original one. Some improvements to elimination methods of Wales are given. The main contribution of the paper is to present an alternative, namely a method to aggregate groups of nodes to produce a factor. The method can be iterated to make hierarchical aggregation schemes. The potential benefits are efficient computation, including recomputation to take into account local changes, and insights into the macroscopic behaviour.

Keywords: Markov flow, aggregation methods, clustering, model reduction

1. INTRODUCTION

Aggregation of Markov processes is an old theme, e.g. [AS], and so is the philosophy of hierarchical aggregation [Si]. Yet, many of the schemes of which we are aware (e.g. [St]) do not look as efficient or accurate as they could be.

The idea of aggregation of a Markov process is to cluster selected nodes into groups and devise an effective Markov process between the groups, that preserves aspects of the group-scale dynamics (also known as “lumping” [KS]). Hierarchical aggregation iterates this procedure to make a hierarchy of descriptions of the same system at successively coarser levels.

Potential outcomes are insight into the macroscopic behaviour, and efficient ways to compute quantities for the system, including efficient re-computation if parts of the system are changed.

There is much active research on aggregation methods more broadly (e.g. see [M]), under names like algebraic multigrid [Wa], coarse-graining [Sh], multilevel or multi-scale analysis [EELRV], lumpability or lumping (e.g. in chemical kinetics [LLJ]), modular decomposition [II], agglomerative clustering [FVH], homogenisation theory [Al], model reduction (as this special issue), dimension reduction [Cu], highway hierarchies [SS], recursive block LU decomposition [DHS, GW], hierarchical incomplete LU decomposition [Saad], scaling laws and renormalisation (e.g. the series [M91, M95, CM] on Frenkel-Kontorova models), and upscaling [OZ]. Many of the methods introduce approximations. Nevertheless, highway hierarchies and the well established Kron reduction [DB] for steady AC power flows (also known as Gauss-Rutishauser elimination or Ward equivalent [MBB]) are exact, as are the above cited methods for Frenkel-Kontorova chains, and so is recursive block LU decomposition.

Date: January 6, 2018.

We say an aggregation method is exact if it produces a factor of the original system. A factor of a system A is a system B with a map h from A onto B such that the image under h of a solution for A is a solution for B . In dynamical systems theory, such a map h is called a semi-conjugacy. The application of h to a solution in other contexts requires interpretation. For example, for Markov flows to be addressed here, the stationary measure at a node of B should be the stationary measure of its pre-image in A .

Presented here is an exact method for aggregation of Markov flows, which can in principle be iterated hierarchically. After writing the paper, we became aware of the stochastic complementation method of [Me], which is closely related but different in some ways. We reserve a comparison until the end.

A *Markov flow* (the terminology appears to go back to [Sg]) is a steady state for a continuous-time regular jump homogeneous semi-Markov process on a discrete state-space V , i.e. a stationary measure with the associated flows and mean first passage times. For the purposes of steady states, the process is characterised by mean waiting time $T_s > 0$ at site s and transition probabilities P_{st} from s to t . One can allow $P_{ss} \neq 0$, or make $P_{ss} = 0$ by dividing T_s and P_{st} by $1 - P_{ss}$; also T_s may be infinite in which case P_{st} do not have meaning. Throughout this paper we shall represent a semi-Markov process by a weighted graph $G = (V, E)$ with weight T_s at node $s \in V$ and weight P_{st} at edge $(s, t) \in E$. If a pair $(s, t) \notin E$ then $P_{st} = 0$. An equivalent formulation is by transition rates $q_{st} = P_{st}/T_s$. ‘‘Semi-Markov’’ means the waiting-time distribution is not assumed to be exponential; all that matters for the steady states is its mean.

We allow sources and sinks. A source is a node s with an external source rate q_s at which matter is added. A sink is a node where all arriving matter is eliminated. Note that the total mass in the system need not be 1. Formally, sources and sinks can be incorporated into the case without them by adding an extra node n with $P_{ns} = q_s / \sum_{r \in R} q_r$ for sources s (with R being the set of sources), $T_n = 1$, and $P_{tn} = 1$ for sinks t (T_t is arbitrary, but say 1), and multiplying the resulting probabilities by $\sum_{r \in R} q_r / \pi_n$, where π_n is the stationary probability of being at n . Alternatively, for source rate 1, the occupation at the source is just the mean first return time. For a survey of computational methods for mean first passage time, albeit in the discrete time context, see [Hu1].

The aim of aggregation methods is to cluster selected nodes into groups to produce a smaller Markov flow with equivalent stationary probability distribution (i.e. the probability of being in a group is the sum of the probabilities of being in its nodes), equivalent stationary fluxes (i.e. the probability flux along an edge of the aggregated system is the sum of the fluxes along the edges of the original system that it represents), and equivalent mean first passage times.

Following [AS], there are schemes to do this, but requiring a global computation for each reduction (e.g. [St]). This strikes us as inefficient.

There are two local schemes for reduction of Markov flows of which we are aware [W1, W2], but they are elimination methods rather than aggregation, and furthermore they work by eliminating one site at a time, so are hierarchical in only a limited sense. We shall review them and give a few refinements, in particular to allow simultaneous elimination of a set of nodes and computation of the stationary probability distribution.

Then we present a local aggregation scheme. The only price to pay is that it requires viewing the Markov flow as having the edges as the nodes, but having made this change of viewpoint, successive aggregation produces a Markov flow on the set of remaining edges.

Formally, what we are aggregating is nodes of a semi-Markov process on edges but we refer to it as aggregation of the associated Markov flow.

We end this introduction by a comparison with lumping methods in monomolecular chemical kinetics. Monomolecular kinetics is Markov dynamics. A starting point for this is [WK]. They aggregate chemical species into pseudo-species in such a way as to make the resulting dynamics close to a factor of the original dynamics. Exact lumpability of the dynamics is extremely rare, however. On the other hand, part of the point of our work is that one can always do exact lumping of the steady state for a system with sources and sinks, which for many purposes may suffice, including those from the petroleum industry motivating [WK]. Secondly, they state that lumping always leads to loss of information. A second point of our work, however, is that it is easy to keep track of how to recover the information about the steady states of the original system. Work on lumping in chemical kinetics has continued, with [LR, LLJ] being a sample of references, but the above two points do not appear to have been appreciated yet. The ideas have also been extended to ecology [A+] and to cell dynamics [B+], but again concentrating on the dynamics rather than steady states.

2. WALES' ELIMINATION METHODS

Wales' methods eliminate nodes one at a time. He uses physicists' notation P_{ts} for P_{st} , so we translate his results to the probabilists' convention, which we consider preferable because we read P_{st} as the transition probability from s to t (represent probability distributions by row instead of column vectors and make transition matrices act on probability distributions to the left instead of the right).

The first method [W1] (called "graph transformation") assumes the convention that $P_{ss} = 0$ for all $s \in V$. To eliminate node x , put

$$(1) \quad \begin{aligned} T'_s &= \frac{T_s + P_{sx}T_x}{1 - P_{sx}P_{xs}}, \\ P'_{st} &= \frac{P_{st} + P_{sx}P_{xt}}{1 - P_{sx}P_{xs}}, \end{aligned}$$

for $s \neq x$ and $t \neq x, s$. It requires examination of only those s from which it is possible to jump to x and those t which can be jumped to from s or x . He proved that for any initial $s \neq x$, if B is a subset not containing x nor s , and the probability of eventually hitting B from s is 1, then the new Markov process computes the correct mean first passage time T_{sB} from s to the set B .

This can facilitate computation. For example, if one successively eliminates all nodes except s and those in B , then T_{sB} is the resulting T'_s . Alternatively, one can eliminate all but B and a small set A containing s and compute the column vector T_{AB} of mean first passage times from $a \in A$ to the set B by the standard formula

$$T_{AB} = (I - P'_{AA})^{-1}T'_A$$

(e.g. [KS], or just note that it satisfies $T_{aB} = T'_a + \sum_{a' \in A} P'_{aa'} T_{a'B}$). A recursive re-grouping method and a recursive enumeration algorithm were used in [CW] to guide the choice of order in which to eliminate nodes.

Wales also produced a second method (“new graph transformation”), that allows $P_{ss} \neq 0$ [W2]. It is preferable in several respects. For example, it conserves T_{sB} even for B containing s . To eliminate node x , put

$$(2) \quad \begin{aligned} T'_s &= T_s + \frac{P_{sx} T_x}{1 - P_{xx}} \\ P'_{st} &= P_{st} + \frac{P_{sx} P_{xt}}{1 - P_{xx}}, \end{aligned}$$

for $s, t \neq x$, including $t = s$. Even if $P_{ss} = 0$ initially, the scheme may produce $P'_{ss} \neq 0$. Computational superiority of this method over a suite of sparse LU solvers was demonstrated in [SW].

In addition, Wales showed how his schemes improve the computation of committor probabilities. This concept applies to the case of a Markov process with more than one communicating component of the set of recurrent nodes. The committor probabilities starting in a node s are the probabilities for the communicating components into which the path is absorbed.

One feature that was not obtained in Wales’ papers is the stationary probability distribution for the new system (uniquely defined in the case that the recurrent set has a unique communicating component), as it was not of particular interest in his applications. It could be useful in other problems, however. This can be done relatively easily. Here is one way (another is given in section 5).

Taking Wales’ second scheme because it allows a little more generality, to compute the stationary probability π_s at node s for the original Markov process from the Markov process after elimination of all but s and a subset B , let T_{ss} be the mean time of first return to s (counting the self-edge ss as a return). Then $T_{ss} = T'_s + \sum_{b \in B} P'_{sb} T_{bs}$ and $T_{bs} = T'_b + \sum_{b' \in B} P'_{bb'} T_{b's}$, so

$$T_{ss} = T'_s + P'_{sB} (I - P'_{BB})^{-1} T'_B,$$

where P'_{sB} is the row vector of P'_{sb} for $b \in B$, P'_{BB} is the part of the new matrix of transition probabilities between nodes of B , and T'_B is the new vector of mean waiting times in nodes of B (T_{ss} may be infinite if for example B contains an absorbing node). Then

$$\pi_s = T_s / T_{ss}.$$

From this, one can also obtain the stationary fluxes ϕ_{sb} from s to $b \in B$ and ϕ_{bs} from $b \in B$ to s , as $\phi_{sb} = \pi_s P_{sb} / T_s$, $\phi_{bs} = \pi_b P_{bs} / T_b$.

Wales’ methods strike us as variants of Kron reduction, which has been used since 1939 in electrical circuit analysis (e.g. [DB]). The context there is undirected weighted graphs, whereas Wales addressed directed weighted graphs. Kron reduction has the advantage over Wales’ methods that one can eliminate any number of nodes simultaneously, subject to inversion of a corresponding matrix.

Actually, the same extension can be made to Wales’ methods, as we now show for his second method.

Definition 1 (Extended new graph transformation). *Given a Markov flow specified by a weighted directed graph $G = (V, E)$ with each node $s \in V$ having a mean waiting time T_s and each edge $(s, t) \in E$ having a transition probability P_{st} associated to it, to eliminate a subset $X \subset V$ of nodes, update the mean waiting time of nodes $s \notin X$ by*

$$(3) \quad T'_s = T_s + P_{sX}(I - P_{XX})^{-1}T_X,$$

and update the transition probability from $s \notin X$ to $t \notin X$ by

$$(4) \quad P'_{st} = P_{st} + P_{sX}(I - P_{XX})^{-1}P_{Xt}.$$

Note that $I - P_{XX}$ is invertible if there is positive probability to eventually leave X from each of its nodes, and by standard arguments for Markov processes this implies probability 1 to leave from each of its nodes. Note also that if there is zero probability of jumping from s straight into X then the waiting time at s remains unchanged. Similarly, the transition probability from s to t remains unchanged if there is zero probability of jumping from s straight into X or from somewhere in X straight to t .

The above scheme is justified by requiring that the mean waiting time T'_s at node $s \notin X$ for the new system be

$$(5) \quad T'_s = T_s + \sum_{x \in X} P_{sx}\bar{T}_x,$$

where \bar{T}_x is the mean time to first escape from X starting in x . Thus $\bar{T}_x = T_x + \sum_{x' \in X} P_{xx'}\bar{T}_{x'}$, which yields the vector $\bar{T}_X = (I - P_{XX})^{-1}T_X$. Inserting this into (5) we obtain equation (3). Similarly, if we require that

$$(6) \quad P'_{st} = P_{st} + \sum_{x \in X} P_{sx}\bar{P}_{xt},$$

where \bar{P}_{xt} is the probability that the first exit from X starting from x is to t , then since $\bar{P}_{xt} = P_{xt} + \sum_{x' \in X} P_{xx'}\bar{P}_{x't}$ we find that $\bar{P}_{Xt} = (I - P_{XX})^{-1}P_{Xt}$. As before, inserting this into (6) we obtain equation (4).

The equations (5, 6) show that T'_s is positive for each s and P'_{st} is positive for each permissible transition. Furthermore, summing (6) over $t \notin X$ shows that $\sum_{t \notin X} P'_{st} = 1$. So a system of the same form as before is obtained.

Just as in Wales' scheme, this extended scheme has the crucial property that upon eliminating a collection of nodes X the mean first passage time from a node $s \notin X$ to a set B disjoint from X remains unchanged.

To implement this extension requires inverting a matrix of size $|X|$, but it might be a useful shortcut in some circumstances. Note that in the case that there are no direct edges between nodes of X , the formulae (3,4) reduce to applying (2) to the nodes of X , in any order, as one should expect.

Wales' methods also strike us as very similar to that of [Sh1, Sh2, Hu2] and the relation is worth exploring.

3. NEW AGGREGATION SCHEME

The main point of this paper is to present a new aggregation scheme, as opposed to an elimination scheme, which requires only local computations and is exact, in the sense of producing a factor of the original system.

Choose a set $A \subset V$ of nodes that we wish to aggregate to one super-node, which we denote again by A . We will achieve this at the expense of possibly (i) introducing self-edges and multiple edges from or to nodes outside A , (ii) making the mean waiting time in A depend on the edge by which A is entered, and (iii) making the transition probabilities from A depend on the entry edge as well as the exit edge.

Formally, this just means the new system is a Markov flow between edges instead of between nodes, i.e. on its line-graph (the line-graph of a graph G is the graph G' whose nodes are the edges of G and which have an edge from e to f if the end-node of e in G is the start-node of f). A Markov flow between nodes can trivially be considered as between its edges. Simply assign to each edge the waiting time at its destination node and make the transition probability from an edge e entering a node s to an edge f leaving s be the probability of moving along f from s to the destination of f (alternatively one could assign to each edge the waiting time of the origin node, but we make the preceding choice). If e and f do not lie end to end then the transition probability between them is zero. We shall call this Markov flow on G' the induced Markov flow on the line-graph. When convenient we will sometimes also like to refer to the Markov flow on edges of G . This is exactly the same as the induced Markov flow on the line-graph, however we think of it as a Markov flow on G but where the states are the edges of G rather than the nodes.

The aggregation scheme to be presented has the feature that if one starts with a Markov flow between edges one obtains a Markov flow between edges. The edges between nodes of A are shrunk to zero, eliminating those edges and aggregating the nodes. We will say an edge is in A if it connects two nodes of A . We say a set A of nodes is *leaky* if there is positive probability from any edge of A to eventually leave A .

Definition 2 (Aggregation scheme). *Given a Markov flow specified by a weighted directed graph $G = (V, E)$, waiting times T_e on edges and transition probabilities P_{ef} between edges, aggregation of a subset $A \subset V$ of nodes is defined by performing the following updates for edges $e, f \in E$ minus those between nodes of A :*

$$(7) \quad \begin{aligned} T'_e &= T_e + P_{eA}(I - P_{AA})^{-1}T_A \\ P'_{ef} &= P_{ef} + P_{eA}(I - P_{AA})^{-1}P_{Af}. \end{aligned}$$

Here P_{eA} denotes the row vector of P_{eu} for edges $u \in A$, P_{AA} is the part of the transition matrix for transitions between edges in A , T_A is the column vector of mean waiting times on edges in A , and P_{Af} is the column vector of P_{uf} for edges $u \in A$. Note that similarly to elimination, the update of mean waiting time need only be performed when e is entering A and the update of transition probability need only be performed when e is entering A and f is leaving A .

An illustration of the aggregation scheme is given in Figure 1. Nodes 5, 6 and 7 are aggregated into a supernode, and edges e, f and g are eliminated. The illustration can be considered part of a larger graph with other edges into and out of nodes 1,2,3 and 4, but no additional ones into or out of nodes 5,6 and 7. For components in order e, f, g , the matrix P_{AA} in this case is given by

$$P_{AA} = \begin{bmatrix} 0 & 0 & P_{eg} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

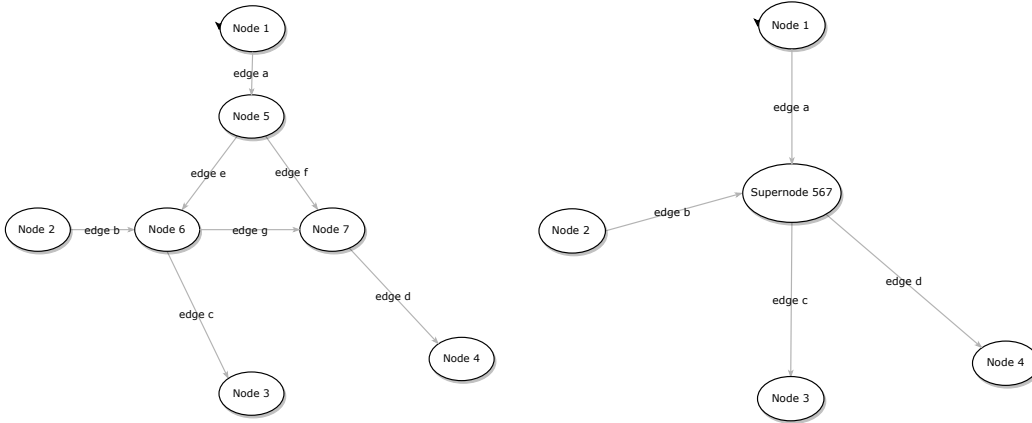


FIGURE 1. Part of a graph and the result of aggregating nodes 5,6,7.

So

$$(I - P_{AA})^{-1} = \begin{bmatrix} 1 & 0 & P_{eg} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Equation (7) tells us how to update the waiting times on edges a, b and the transition probabilities from a, b to c, d .

The derivation of (7) is very similar to that of (3) and (4) for elimination of a group of nodes. In detail, the mean waiting time T'_e on edge e entering A for the new system has to take into account the mean time to be spent in A before exiting. Thus $T'_e = T_e + \sum_{u \in A} P_{eu} \bar{T}_u$, where \bar{T}_u is the mean time to first escape from A starting on u . Now $\bar{T}_u = T_u + \sum_{v \in A} P_{uv} \bar{T}_v$, so the vector $\bar{T}_A = (I - P_{AA})^{-1} T_A$. Thus $T'_e = T_e + P_{eA} (I - P_{AA})^{-1} T_A$. Similarly, $P'_{ef} = P_{ef} + \sum_{u \in A} P_{eu} \bar{P}_{uf}$, where \bar{P}_{uf} is the probability that the first exit from A starting on edge $u \in A$ is to edge f . This satisfies the equation $\bar{P}_{uf} = P_{uf} + \sum_{v \in A} P_{uv} \bar{P}_{vf}$, so the vector $\bar{P}_{Af} = (I - P_{AA})^{-1} P_{Af}$. Thus $P'_{ef} = P_{ef} + P_{eA} (I - P_{AA})^{-1} P_{Af}$. Note that $(I - P_{AA})$ is invertible under the condition that A is leaky. If A is not leaky there is no great point in computing when and to where one exits A . Note also that one can check that $\sum_f P'_{ef} = 1$ for the new system.

A downside of the aggregation scheme is that it may generate multiple edges between two nodes of the new system. For example, if there are edges from a node s not in A to two nodes in A , the new system will have two edges from s to A . Similarly, if there are edges from two nodes in A to a node t not in A , the new system will have two edges from A to t . One might hope that multiple edges could be aggregated into single edges. In general, however, this is not possible: if edges f_i from a node s to a node t are to be aggregated into a single edge f then for every edge e into s we must have $P_{ef} = \sum_i P_{ef_i}$, and for all edges e into s and g from t we must have $\sum_i P_{ef_i} P_{f_i g} = P_{ef} P_{fg}$, but if there is more than one edge e into s this gives incompatible assignments to P_{fg} in general. One solution is to aggregate s and t into one node, thereby not aggregating the edges f_i but eliminating them. This may produce other multiple edges elsewhere, however. Probably one should take care not to generate too many multiple edges in the choice of

which nodes to aggregate. The scheme may also generate self-edges, but as with Wales' second method, this is not necessarily a disadvantage.

Further aggregation can be performed, including putting super-nodes into clusters. They have no different status from the original nodes. One can also perform aggregation of disjoint subsets simultaneously, as the updates for edges entering or leaving disjoint sets of nodes do not interact.

At any stage of aggregation, the mean first passage time from an edge e to a set B of edges not containing e is given by solving the usual system of equations

$$T_{eB} = T_e + \sum_{f \notin B} P_{ef} T_{fB}$$

for the vector of mean first passage times T_{eB} .

One can think of the mean first passage time from a node s to a node t as being the occupation density at s resulting from injecting flux at rate 1 at s and taking out all matter arriving at t . Thus it corresponds to the simplest case of a source and sink. We can tackle multiple sources and sinks, however, along the lines indicated in the Introduction.

The aggregation scheme can also be used to compute stationary probabilities. For example, the fraction on edge e is T_e/T_{ee} where T_{ee} is the mean time of first return to e , whose computation can be done via aggregation. Alternatively, the procedure of section 5 can be applied.

4. AGGREGATION OF NODES AS ELIMINATION OF EDGES

It was mentioned in the previous section that the new aggregation method eliminates the edges between the nodes to be aggregated and that the derivation of (7) is very similar to that of (3,4). Actually, the new method is precisely elimination of nodes in the line-graph, using the version (3,4). This is made precise in the following proposition. It shall sometimes be convenient to include the weights explicitly by writing a quadruple $G = (V, E, T, P)$ to characterise our Markov flow.

Proposition 1. *Let $G = (V, E, T, P)$ be a Markov flow, and let $X \subset V$. If one aggregates X then one obtains a Markov flow on edges which naturally induces a Markov flow $G^a = (V^a, E^a, T^a, P^a)$ on the line-graph. Alternatively one may first naturally induce a Markov flow on the line-graph $G' = (V(G'), E(G'))$ of G . Let B be the set of edges in X . If one eliminates the collection of nodes in G' induced by the edges B in G then one obtains a new Markov flow $G^e = (V^e, E^e, T^e, P^e)$. The graphs G^a and G^e are isomorphic.*

The proposition is illustrated in Figure 2.

First let us set up some notation. Let $G_X = (V_X, E_X)$ denote the weighted directed graph resulting from aggregating X . Also, the standard procedure for constructing G' from G yields a bijection $L : E \rightarrow V(G')$ (L for "line-graph"), which we call the natural bijection from edges of G to nodes of G' . When talking about any graph, not necessarily G , we shall also refer to the natural bijection from the edges of the original graph to the nodes of its line-graph.

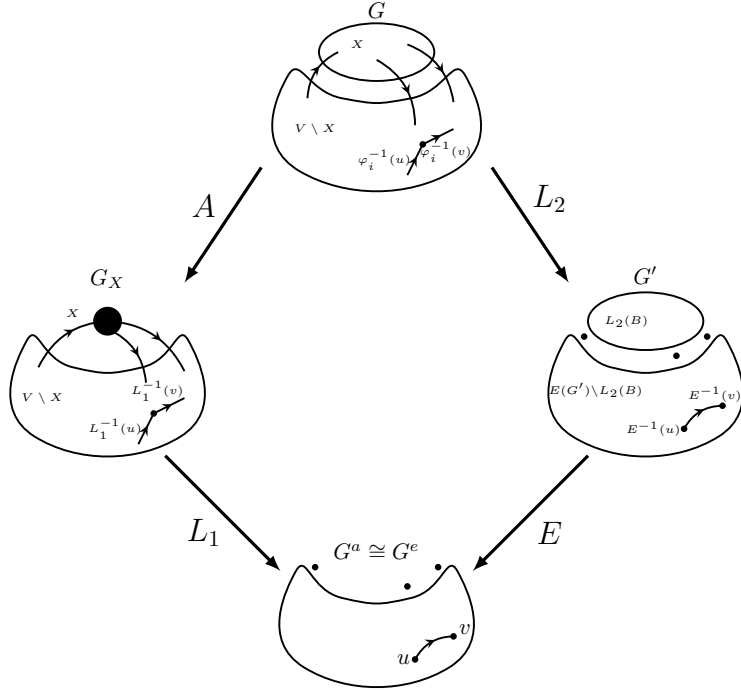


FIGURE 2. Illustration of Proposition 1

Proof of Proposition 1. We will show that there is an isomorphism φ between (V^a, E^a) and (V^e, E^e) as directed graphs without weights. Then we show φ preserves weights too. It is worth noting at this point that even if G were a multigraph (i.e. with possibly more than one directed edge from a node to another node), both G^a and G^e are merely graphs, and not multigraphs (this is a symptom of the fact that construction of both G^a and G^e involves taking a line-graph, and not augmenting the set of edges after this action). So when showing our claimed isomorphism is indeed an isomorphism we know there can be at most a single edge between any two nodes.

First we find a bijection between $E \setminus B$ and V^a , as follows. Upon aggregation of X in G the edges that are eliminated are exactly the edges in X , that is B . So there is a bijection $A : E \setminus B \rightarrow E_X$ (A for “aggregation”). Upon taking the line-graph of G_X we have a natural bijection (defined above) $L_1 : E_X \rightarrow V^a$, thus yielding the bijection $\varphi_1 = L_1 \circ A$ we sought.

Similarly we construct a bijection between $E \setminus B$ and V^e . Let $L_2(B)$ denote the collection of nodes in G' induced by edges B . There is a natural bijection $L_2 : E \setminus B \rightarrow V(G') \setminus L_2(B)$. Then as we eliminate exactly the set $L_2(B)$ of nodes in the line-graph to form G^e there is trivially a bijection $E : V(G') \setminus L_2(B) \rightarrow V^e$ (E for “elimination”) and thus we find a bijection $\varphi_2 = E \circ L_2$. Hence $\varphi = \varphi_2 \circ \varphi_1^{-1}$ is a bijection $V^a \rightarrow V^e$.

Now we show that this bijection is indeed an isomorphism. Let $u, v \in V^a$. We will show that $(u, v) \in E^a$ if and only if either (i) $\varphi_1^{-1}(u)$ is an edge entering X and $\varphi_1^{-1}(v)$ is an edge leaving X , or (ii) the edges $\varphi_1^{-1}(u)$ and $\varphi_1^{-1}(v)$ in G lie end to end with the

destination node of $\varphi_1^{-1}(u)$ being the origin of $\varphi_1^{-1}(v)$. We shall also prove the analogous statement but taking instead $u, v \in V^e$ and replacing φ_1 with φ_2 in the above. These two facts together imply φ is an isomorphism.

Suppose $(u, v) \in E^a$, then certainly the edges $L_1^{-1}(u)$ and $L_1^{-1}(v)$ in E_X lie end to end with the destination s of $L_1^{-1}(u)$ equalling the origin of $L_1^{-1}(v)$. We consider two cases. First suppose that $s \neq X$ (remember, when we aggregate a collection of nodes X in G we denote by X also the supernode in G_X). Then, upon undoing the aggregation, the nodes $\varphi_1^{-1}(u) = A^{-1} \circ L_1^{-1}(u)$ and $\varphi_1^{-1}(v) = A^{-1} \circ L_1^{-1}(v)$ still lie end to end. In the case $s = X$ then clearly $\varphi_1^{-1}(u) = A^{-1} \circ L_1^{-1}(u)$ is an edge entering X and $\varphi_1^{-1}(v) = A^{-1} \circ L_1^{-1}(v)$ is an edge leaving X since this is the only way that the destination of $L_1^{-1}(u)$ and origin of $L_1^{-1}(v)$ are both X .

Conversely suppose u, v are such that either (i) or (ii) holds. If we are in case (i) then the destination of $L_1^{-1}(u) = A \circ \varphi_1^{-1}(u)$ and the origin of $L_1^{-1}(v) = A \circ \varphi_1^{-1}(v)$ are both X . Since this means they lie end to end, upon taking the line-graph we recover that $(u, v) \in E^a$. If we are in case (ii) then again the destination of $L_1^{-1}(u)$ and the origin of $L_1^{-1}(v)$ must coincide (but not be equal to X , of course). Just as before, this implies that $(u, v) \in E^a$. Through highly similar arguments one can prove the corresponding statement originally taking instead nodes $u, v \in E^e$ and replacing φ_1 with φ_2 , thus proving that φ is an isomorphism.

The final thing to check is that $T_u^a = T_{\varphi(u)}^e$ and $P_{uv}^a = P_{\varphi(u)\varphi(v)}^e$ for all $u, v \in V^a$. This is almost trivial to see once one notices that when one naturally induces a Markov flow on edges of G , the mean waiting times and transition probabilities are just $T_e = T_{L_2(e)}$ and $P_{ef} = P_{L_2(e)L_2(f)}$ for all $e, f \in E$. So finally, comparing equations (3,4) and (7) we see the updates are identical. \square

The main conclusion to draw here is that to perform aggregation one can instead transfer to the line-graph, perform elimination and then undo the line-graph transformation. In a loose way one can think of this as meaning that aggregation and elimination are conjugate in the sense that $LA = EL$ where A, E, L represent performing aggregation, elimination, and moving to the line-graph respectively.

5. RELATION TO BLOCK LU DECOMPOSITION

Another significant observation is that elimination schemes for Markov flows, whether on a graph or a line-graph, are closely related to block LU decomposition, as defined in [DHS, GW] for example. In particular, this connection makes clear how to keep track of quantities for parts for the system after they have been eliminated.

Let us review block LU decomposition, following [DHS]. To solve a linear system $Ax = b$ for x , split the components into two groups $\{1, 2\}$ chosen so that A_{11} is easy to invert. The idea is that we will eliminate the first group of components to reduce the problem to some linear system $A'x_2 = y_2$ but keep enough information to recover x_1 too. First solve

$$L_{21}A_{11} = A_{21}$$

for L_{21} (which is easy by hypothesis), and then write

$$(8) \quad \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A' \end{bmatrix}$$

with

$$A' = A_{22} - L_{21}A_{12}.$$

To solve $Ax = b$ for x , first put

$$y_2 = b_2 - L_{21}b_1,$$

then solve

$$A'x_2 = y_2$$

for x_2 (this involves recursion because A' will in general need simplifying in the same way as we did with A), then solve

$$A_{11}x_1 = b_1 - A_{12}x_2$$

for x_1 , which is easy because we supposed A_{11} is easy to invert.

Next, we apply block LU decomposition to the problem of computing the stationary probability distribution for a semi-Markov process. For non-eliminated nodes, this boils down to an application of Wales' second method, but the block LU procedure shows how to compute it for eliminated nodes too. Suppose mean waiting time T_s at node $s \in V$ and probability P_{st} for the next node t (allow $P_{ss} \neq 0$). Stationary probability π corresponds to $\pi_s = \rho_s T_s$ with $\rho P = \rho$ and $\rho T = 1$. We can solve for ρ by LU decomposing the augmented matrix $[I - P \quad T]$ into parts for two disjoint node sets X and Y where $X \cup Y = V$. X is assumed to be leaky (so $I - P_{XX}$ is invertible) and $I - P_{XX}$ is assumed to be easy to invert. The form of the desired LU decomposition is:

$$(9) \quad \begin{bmatrix} I - P_{XX} & -P_{XY} & T_X \\ -P_{YX} & I - P_{YY} & T_Y \end{bmatrix} = \begin{bmatrix} I & 0 \\ -L_{YX} & I \end{bmatrix} \begin{bmatrix} I - P_{XX} & -P_{XY} & T_X \\ 0 & I - P'_{YY} & T'_Y \end{bmatrix}$$

The block components are found by solving

$$L_{YX}(I - P_{XX}) = P_{YX}$$

for L_{YX} and then putting

$$P'_{YY} = P_{YY} + L_{YX}P_{XY}$$

and

$$T'_Y = T_Y + L_{YX}T_X.$$

To find ρ we solve

$$\rho_Y P'_{YY} = \rho_Y, \quad \rho_Y T'_Y = 1$$

for ρ_Y (recursively in general) and then extend by

$$\rho_X = \rho_Y L_{YX}.$$

At the end we put $\pi_s = \rho_s T_s$.

The same applies to a semi-Markov process on a line-graph. Just use the edges as the nodes. Thus we can compute the stationary probability using our aggregation scheme. If the process on the line-graph came from one on a graph then the stationary probability at a node is given by summing the stationary probabilities on the edges entering that node.

Lastly, we can use the same insight to compute mean first passage times to a subset B of nodes from all nodes outside B , not just non-eliminated ones. The vector of mean first passage times T_a^B from nodes $a \in A = B^c$ to a set of nodes B is the solution of $(I - P_{AA})T_A^B = T_A$. Suppose we eliminate set X of nodes, disjoint from B , and we let C be the complement of $X \cup B$. Use the above LU decomposition, but just the part for C in Y . So solve

$$(I - P'_{CC})T_C^B = T'_C$$

for T_C^B (recursively in general). Then solve

$$(I - P_{XX})T_X^B = T_X + P_{XC}T_C^B$$

for T_X^B . One remaining issue to address is what happens if we eliminated some nodes of B , but we leave that for later investigation.

6. DYNAMICS

Lastly, one might ask whether an aggregation scheme can be devised that keeps the dynamics, not just the steady state. Even in the Markov case, this can not be done with the above schemes because the dynamics of an N -state Markov chain has $N - 1$ eigenvalues (counting multiplicity) so reduction of N must lose some of the dynamics. It can be done, however, if we replace the mean waiting time by the Laplace transform of the distribution of waiting times. This is because each Laplace component solves a steady state problem. The downside is that aggregation introduces rational functions of the Laplace transform variable s , whose degree increases as the number of nodes decreases.

7. CONCLUSION

A promising method for aggregation of Markov flows has been presented, which lends itself to iteration to produce a hierarchy.

After writing the paper we became aware of [Me] on stochastic complementation. Like our method, this is an adaptation of block LU decomposition to stochastic matrices. Meyer's paper treats discrete time, whereas ours treats continuous time, but that is not a great difference. Instead of eliminating part of the system and working out how to keep track of properties for the eliminated part, he divides the system into two parts with equal status and some coupling between them. He treats only stationary distribution whereas we also treat mean first passage times, though again that is not a big extension. The main innovative feature of our paper that we think is useful is to perform the elimination on the line graph rather than the original graph.

The next steps for our aggregation method are to compute mean first passage times using the hierarchy of resolutions, determine useful criteria to choose the nodes to aggregate at each step, and estimate the computational complexity of the resulting hierarchical scheme. A clear criterion for choosing the nodes to aggregate is to choose groups of nodes which have relatively few edges to and from the outside, compared to their internal edges. Heuristics and principles which may be useful include clustering algorithms [FVH, CKMM], community detection [FH, POM], and dissection methods [GBDD]. The literature on methods for sparse linear systems [DRS] will be very useful too.

A subsequent paper will report on tests of the method, including choices of hierarchy. For instance, we plan to compare with benchmark examples in [SW] on configurational transitions for Lennard-Jones clusters and a β -sheet peptide.

COMPETING INTERESTS

We have no competing interests.

AUTHORS' CONTRIBUTIONS

RM devised the aggregation scheme. JR showed that it is exactly an elimination scheme on the line-graph. We both worked out the application of block LU decomposition to computing stationary probabilities. JR wrote the proof that the aggregation scheme is an elimination scheme on the line-graph. RM wrote the rest with some polishing by JR. Both authors gave final approval for publication.

ACKNOWLEDGEMENTS

RM is grateful to David Wales for stimulating his interest in this topic. The ideas here were developed in response to Wales' papers. The material of sections 1–3 was taught in a postgraduate module in Spring 2012 and RM is grateful to Erasmus Mundus masters students Aditya Challa and Yan Zhang for writing a report about them. The writing was carried out under a Royal Society Wolfson Research Merit Award for RM, followed by final year MMath project of JR. We are grateful to David Bindel for bringing stochastic complementation and nested dissection to our attention, and to a reviewer for references on lumping in chemical kinetics.

FUNDING STATEMENT

RM was supported by the University of Warwick and a Royal Society Wolfson Research Merit Award.

REFERENCES

- [Al] Allaire G, 2012, Introduction to homogenisation theory, in: *Mathematical and numerical approaches for multi-scale problems*, eds Cancès E, Labbé S, ch.1
- [AS] Ando A, Simon HA, 1961, Aggregation of variables in dynamical systems, *Econometrica* 29, 111–138.
- [A+] Auger P, de la Para RB, Poggiale JC, Sanchez E, Huu TN, 2008, Aggregation of variables and applications to population dynamics, in: *Structured Population Models in Biology and Epidemiology*, LNM 1936, *Mathematical Biosciences Subseries*. Edited by: Magal P, Ruan S. pp 209-263. Berlin: Springer
- [B+] Bérenguier D, Chaouiya C, Monteiro PT, Naldi A, Remy E, Thieffry D, Tichit L. 2013, Dynamical modeling and analysis of large cellular regulatory networks. *Chaos* 23(2):025114.
- [CW] Carr JM, Wales DJ, 2008, Folding pathways and rates for the three-stranded β -sheet peptide Beta3s using discrete path sampling, *J Phys Chem B* 112, 8760–9
- [CM] Catarino N, MacKay RS, 2005, Renormalization and quantum scaling of Frenkel-Kontorova models, *J Stat Phys* 121, 995–1014.
- [CKMM] Cohen-Addad V, Kanade V, Mallmann-Trenn F, Mathieu C, 2017, Hierarchical Clustering: Objective Functions and Algorithms, arXiv: 1704.02147
- [Cu] Cunningham P, 2007, Dimension reduction, Univ Coll Dublin tech rpt UCD-CSI-2007-7

- [DRS] Davis TA, Rajamanickam S, Sid-Lakhdar WM, 2016, A survey of direct methods for sparse linear systems, *Acta Num* 383–566
- [DHS] Demmel JW, Higham NJ, Schreiber RS, 1995, Stability of block LU factorization, *Num Lin Alg Appl* 2, 173–190
- [DB] Dörfler F, Bullo F, 2013, Kron reduction of graphs with applications to electrical networks, *IEEE Trans Circuits Sys I* 60, 150–163.
- [EELRV] E W, Engquist B, Li X, Ren W, Vanden-Eijnden E, 2007, Heterogeneous multiscale methods: a review, *Comm Comput Phys* 2, 367–450.
- [FH] Fortunato S, Hric D, 2016, Community detection in networks: a user guide, *Phys Rpts* 659, 1–44.
- [FVH] Fränti P, Virtajoki O, Hautamäki V, 2006, Fast agglomerative clustering using a k -nearest neighbour graph, *IEEE Trans Pattern & Machine Intell* 28, 1875–81.
- [GW] Georgiev K, Wasniewski J, 2001, Recursive version of LU decomposition, in: Vulkov L, Wasniewski J, Yalamov P (eds), *Lect Notes Comp Sci* 1988 (Springer) 325–32
- [GBDD] Grigori L, Boman EG, Donfack S, Davis TA, 2010, Hypergraph-based unsymmetric nested dissection ordering for sparse LU factorization, *SIAM J Sci Comput* 32, 3426–46
- [Hu1] Hunter JJ, 2017, The computation of mean first passage times for Markov chains, arXiv 1701.07781
- [Hu2] Hunter JJ, 2016, Accurate calculations of Stationary Distributions and Mean First Passage Times in Markov Renewal Processes and Markov Chains, *Special Matrices* 4, 151–175
- [II] Ille P, 1997, Indecomposable graphs, *Discrete Math* 173, 71–8
- [KS] Kemeny JG, Snell JL, 1976, *Finite Markov chains* (Springer)
- [LR] Li, G., Rabitz, H., 1989. A general analysis of exact lumping in chemical kinetics. *Chem. Eng. Sci.* 44, 14131430.
- [LLJ] Lin, B., Leibovici, C.F., Jorgensen, S.B., 2008. Optimal component lumping: problem formulation and solution techniques. *Comput. Chem. Eng.* 32, 1167–72
- [MBB] Machowski J, Bialek JW, Bumby JR, 2008, *Power system dynamics* (Wiley).
- [M91] MacKay RS, 1991, Scaling exponents at the transition by breaking of analyticity for incommensurate structures, *Physica D* 50, 71–79
- [M95] MacKay RS, 1995, The classical statistical mechanics of Frenkel-Kontorova models, *J Stat Phys* 80, 45–67
- [M] MacKay RS, 2011, Hierarchical aggregation of complex systems, in: CRP2011 Position Papers, http://lora.maths.warwick.ac.uk/groups/eccs_sc/wiki/c8c73/CRP_Forum_2011.html (for EC FET-Proactive call on “Dynamics of multilevel complex systems”, session at ECCS11, Vienna).
- [Me] Meyer CD, 1989, Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems, *SIAM Rev* 31, 240–272
- [OZ] Owhadi H, Zhang L, 2007, Metric-based upscaling, *Commun Pure Appl Math* 60, 675–723.
- [POM] Porter MA, Onnela J-P, Mucha PJ, 2009, Communities in networks, *Notice Am Math Soc* 56, 1082-97, 1164-6.
- [Saad] Saad Y, www-users.cs.umn.edu/~saad/software
- [SS] Sanders P, Schultes D, 2006, Engineering highway hierarchies, in: *ESA 2006, Lect Notes in Computer Science* 4168 (Springer) 804-816
- [Si] Simon HA, 1962, The Architecture of Complexity, *Proc Am Phil Soc* 106, 467-482.
- [Sg] Sgurev V, 1993, Markov flows (Bulgarian Academy of Sciences, Sofia) (in Russian)
- [Sh] Shell MS, 2016, Coarse-graining with the relative entropy, in: *Advances in Chemical Physics*, eds Rice SA, Dinner AR (Wiley) ch.5
- [Sh1] Sheskin TJ, 1985, A Markov partitioning algorithm for computing steady state probabilities. *Oper. Res.* 33, 228-235
- [Sh2] Sheskin TJ, 1995, Computing mean first passage times for a Markov chain, *International Journal of Mathematical Education in Science and Technology*, 26:5, 729-735
- [SW] Stevenson JD, Wales DJ, 2014, Analysing kinetic transition networks for rare events, *J Chem Phys Commun* 141, 041104

- [St] Stewart WJ, 1994, Introduction to the numerical solution of Markov chains (Princeton)
- [Wa] Wagner AMG, 1998/9, Introduction to algebraic multigrid, Lecture notes, Heidelberg
- [W1] Wales DJ, 2006, Energy landscapes: calculating pathways and rates, Int Rev Phys Chem 25, 237-282.
- [W2] Wales DJ, 2009, Calculating rate constants and committor probabilities for transition networks by graph transformation, J Chem Phys 130, 204111
- [WK] Wei, J., Kuo, J.C., 1969. A lumping analysis in monomolecular reaction systems: analysis of the exactly lumpable system. Ind. Eng. Chem. Fundam. 8, 114123

MATHEMATICS INSTITUTE AND CENTRE FOR COMPLEXITY SCIENCE, UNIVERSITY OF WARWICK,
COVENTRY CV4 7AL, UK

E-mail address: R.S.MacKay@warwick.ac.uk and J.D.Robinson@warwick.ac.uk