16th World Congress in Computational Mechanics (WCCM)
21-26 July 2024, Vancouver, Canada
A. Korobenko, M. Laforest, S. Prudhomme, R. Vaziri (Eds)

A PROBABILISTIC GRAPHICAL MODEL APPROACH TO INTERPRET, VERIFY, AND DECOUPLE MULTI-PHYSICS SYSTEMS

TEO PRICE-BRONCUCIA[1], SIENNA AMORESE[1*], RICARDO BAPTISTA[2] AND REBECCA MORRISON[1†]

[1] University of Colorado Boulder
1111 Engineering Dr, Boulder, CO, USA 80309
teo.pricebroncucia@colorado.edu; * sienna.amorese@colorado.edu; † rebeccam@colorado.edu

[2] California Institute of Technology
1200 E California Blvd, Pasadena, CA, USA 91125
rsb@caltech.edu

Key words: Undirected Graphical Models, Conditional Independence, Numerical Models.

Abstract. Multi-physics scientific codes, like those used in weather prediction and spacecraft simulation, often involve the coupling of many subdomain models. The resulting models are generally expensive to run. These costs, along with the model's complexity, make downstream tasks like model calibration and uncertainty quantification especially difficult. In this work, we simplify structure in multi-physics models by learning an undirected graphical model corresponding to the system state variables, where the edges in the learned graph represent conditional dependence between variables. Depending on the application of interest, the resulting graph may (1) reveal the probabilistic structure of the joint distribution; (2) identify the most important coupling variables (those that must be shared between subdomain models), and those which may be safely neglected; (3) identify candidate variables for first-pass verification tasks; or (4) decouple parts of a model, while maintaining accuracy in the model predictions. We illustrate these possibilities through two multi-physics numerical models, the Multiple Prediction Across Scales-Atmosphere (MPAS-A) code base and a fire detection satellite model.

1 INTRODUCTION

Probabilistic graphical models (PGMs) are simple but powerful objects to organize and simplify relationships among random variables. Fundamental to many fields such as statistical physics, machine learning, and natural language processing, they have more recently been lever-

aged in computational science and engineering applications. Notably, [1] laid a PGM foundation to accelerate inference and prediction for large-scale digital twins. Along this trend, we introduce PGMs to identify and exploit sparse structure in coupled, multidisciplinary systems.

Coupled computation regularly occurs in realistic engineering applications. An aerospace system model may integrate structural designs, fluid mechanics, thermal processes, etc. Climate models couple land, sea, ice, soil, precipitation, and many other domains. Such coupling may entail expensive feedback (iterative solutions), heavy communication overhead, or delicate consistency of scenario parameters across domains [2]. Moreover, inputs, coupling variables, and quantities of interest (QoIs) are often characterized as random variables (due to several possible sources of uncertainty) leading to even higher computational expense if the coupling needs to iteratively solve for many random inputs. Thus, a goal of great interest is to identify and exploit simplified or decoupled model structure.

In this paper, we leverage theory and techniques from the field of PGMs to simplify model structure, perform efficient code verification tasks, and learn possible decouplings. First we associate random system state variables with nodes of the graph. Given data from either observations of the true physical system, or a high-fidelity model, we then show how to identify the graph edges and their weights. A necessary coupling of the model corresponds to a strong edge of the graph. If the resulting graph is sparse, we thus learn how to safely decouple these complex systems, while still maintaining reliability in the model's predictive performance. Even if subdomain models cannot be decoupled, information about the probabilistic structure may help users select the most effective variables for tasks like first-pass verification.

In many cases, inherent spatial- or time-dependence of a system naturally induces a PGM. For example, consider a standard hidden Markov model of the daily average temperature in Boulder, Colorado and the corresponding observations. Here, we assume the temperature evolves as a Markov process and the measurement errors are independent. This readily implies an undirected graph, in this case a tree, (see Ch. 15 in [3]), a model structure which is especially beneficial for other tasks such as filtering, smoothing, and prediction. But in the case of coupling between outputs or internal variables of a multi-physics model, there may be no time or spatial element (nor any other obvious physical structure) which naturally leads to a sparse graph. Instead, the graph must be either assumed by a domain expert or learned from available data. This data could come from runs of a high-fidelity model or from the physical system itself. Learning a sparse graph in the context of multi-physics computational models is the goal of this paper.

2 SOME THEORY

We consider a general nonlinear system to be given by the coupled set of M equations

$$\begin{aligned} F_1(x_1, \ldots, x_M; z_1) &= 0 \\ &\vdots \\ F_M(x_1, \ldots, x_M; z_M) &= 0, \end{aligned} \tag{1}$$

where $\boldsymbol{x} = \{x_i\}_{i=1}^M$ are the state variables (which includes coupling and output/QoI variables) and $\boldsymbol{z} = \{z_i\}_{i=1}^M$ are random inputs. Given a distribution ρ_0 for $\boldsymbol{z}$, there exists a distribution for the states that solve (1), whose density we denote by π.

An *undirected graphical model* (UGM) $\mathcal{G} = (V, E)$ consists of a set of vertices V and undirected edges E. In a probabilistic UGM, each node corresponds to a random variable, and the edges visually represent conditional dependencies between them. More specifically, the lack of an edge between nodes i and j means that variables X_i and X_j are conditionally independent, as in

$$(i, j) \notin E \iff X_i \perp\!\!\!\perp X_j | X_{\mathcal{X}}, \tag{2}$$

where $\mathcal{X}$ may refer to all variables except X_i and X_j, a neighborhood of X_i or X_j, or a separator set between nodes i and j. These three conditioning sets correspond to the pairwise (P), local (L), and global (G) Markov properties, respectively. In general, $(G) \implies (L) \implies (P)$, but if the joint distribution is positive everywhere, then the three Markov properties are in fact equivalent, i.e., $(P) \implies (G)$. See [4, 3] for more details about undirected (and directed) graphical models.

2.1 Linear Gaussian models

For a multivariate normal distribution, the graph is revealed by inspection of the inverse covariance matrix, or precision. That is, for Gaussian UGMs with joint density π and covariance Σ_π, we have

$$(i, j) \notin E \iff (\Sigma_\pi)^{-1}_{ij} = 0. \tag{3}$$

This brings us to our first lemma, which connects graphical models to multi-physics systems when the coupling variables are Gaussian and the model is linear.

Lemma 2.1 (Linear Gaussian models.)**.** *Let the model be affine, such that the equations in (1) can be written in matrix form as*

$$\boldsymbol{A}\boldsymbol{x} - \boldsymbol{B}\boldsymbol{z} - \boldsymbol{c} = \boldsymbol{0}, \tag{4}$$

where $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{M\times M}$, $\boldsymbol{A}^{-1}$ *exists,* $\boldsymbol{c} \in \mathbb{R}^{M\times 1}$, *and* $\boldsymbol{z} \sim \mathcal{N}(\mu_{\rho_0}, \Sigma_{\rho_0})$. *Then*

$$\Sigma_\pi^{-1} = \boldsymbol{A}^T(\boldsymbol{B}\Sigma_{\rho_0}\boldsymbol{B}^T)^{-1}\boldsymbol{A}.$$

Proof. For this system with invertible $\boldsymbol{A}$, the outputs are a linear function of the inputs $\boldsymbol{z}$: $\boldsymbol{x} = \boldsymbol{A}^{-1}(\boldsymbol{B}\boldsymbol{z}+\mathbf{c})$. In particular, if the inputs follow a Gaussian distribution $\rho_0 = \mathcal{N}(\mu_{\rho_0}, \Sigma_{\rho_0})$, the outputs are also Gaussian with the distribution

$$\pi = \mathcal{N}(\mu_\pi, \Sigma_\pi) = \mathcal{N}(\boldsymbol{A}^{-1}(\boldsymbol{B}\mu_{\rho_0} + \boldsymbol{c}), \boldsymbol{A}^{-1}\boldsymbol{B}\Sigma_{\rho_0}\boldsymbol{B}^T\boldsymbol{A}^{-T}). \tag{5}$$

Inverting the covariance above, the precision matrix is $\Sigma_\pi^{-1} = \boldsymbol{A}^T(\boldsymbol{B}\Sigma_{\rho_0}\boldsymbol{B}^T)^{-1}\boldsymbol{A}$. □

One goal of our work is to use the sparsity of this matrix Σ_π^{-1} to accelerate downstream tasks, such as decoupling, inference and prediction, and code coverage and verification tasks.

Example 2.1. *Let* $M = 5$, *assume that* $\Sigma_{\rho_0} = \boldsymbol{I}$, *and suppose the model is given by*

$$\begin{aligned} x_1 &= b_{11}z_1 + c_1 \\ x_2 &= -a_{21}x_1 + b_{22}z_2 + c_2 \\ x_3 &= b_{33}z_3 + c_3 \\ x_4 &= -a_{43}x_3 + b_{44}z_4 + c_4 \\ x_5 &= -a_{53}x_3 - a_{54}x_4 + b_{55}z_5 + c_5 \end{aligned} \qquad \textit{and thus} \qquad \boldsymbol{A} = \begin{bmatrix} 1 & & & & \\ a_{21} & 1 & & & \\ & & 1 & & \\ & & a_{43} & 1 & \\ & & a_{53} & a_{54} & 1 \end{bmatrix}.$$

Note $\boldsymbol{B}$ *is a diagonal matrix (i.e., each random input only appears in one subdomain model) and* $\boldsymbol{A}$ *is a (lower-triangular) block matrix, which together imply that the first two outputs,* x_1 *and* x_2 *can be decoupled from* x_3, x_4, *and* x_5. *The precision matrix is* $\Sigma_\pi^{-1} = \boldsymbol{A}^T\boldsymbol{D}\boldsymbol{A}$ *where* $d_{ii} = b_{ii}^2$. *Of note,* Σ_π^{-1} *follows the same block structure as* $\boldsymbol{A}$.

The point is that if we did not know the model, but only had access to data of model output, we could numerically compute Σ_π^{-1} to learn the model sparsity. The case study of Section 3.1 will do exactly that.

2.2 Nonlinear or non-Gaussian models

Coupled, nonlinear models often admit non-Gaussian randomness, but learning non-Gaussian graphs is significantly more difficult than Gaussian graphs. In general, we cannot rely on the precision matrix anymore (although for certain non-Gaussian distributions, the precision matrix may still be a good indicator of the graph; see, for example, [5]). Instead, we compute

another measure of conditional independence: For a continuous distribution with differentiable log-density $\pi(\boldsymbol{x})$, where π is positive over $\mathbb{R}^d$, then, as shown in [6],

$$(i,j) \notin E \iff \partial_i \partial_j \log \pi(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{6}$$

In words, there is no edge between X_i and X_j if the ij-th mixed partial derivative of the log density is 0 everywhere. Following [7], the conditional independence score is defined as the matrix Ω, where $\Omega_{ij} \equiv \mathbb{E}_\pi \left[(\partial_i \partial_j \log \pi(\boldsymbol{x}))^2 \right]$. This leads to our second lemma, which connects UGMs to multi-physics systems for non-Gaussian distributions.

Lemma 2.2 (Non-Gaussian distributions, following [7])**.** *Let the complete model be nonlinear, or the inputs $\boldsymbol{z}$ be non-Gaussian, such that $\boldsymbol{x}$ follows a non-Gaussian distribution. In this case, the sparsity of the multi-physics model is given by the zeroes of Ω.*

Proof. A sufficient condition to eliminate edge (i,j) from the graph is that $\Omega_{ij} = 0$. □

Note that for Gaussian π, the result reduces to that in equation (3): $\pi(\boldsymbol{x}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T \Sigma_\pi^{-1} \boldsymbol{x}\right)$ so that $\partial_i \partial_j \log \rho(\boldsymbol{x}) = -(\Sigma_\pi^{-1})_{ij}$.

Example 2.2 (Rosenbrock function)**.** *Let $M = 3$, μ, a, b, c be constants, and let the model be defined as:*

$$\begin{aligned} x_1 &= \epsilon_1, \quad \epsilon_1 \sim \mathcal{N}(\mu, 1/(2a)) \\ x_2 &= \epsilon_2, \quad \epsilon_2 \sim \mathcal{N}(x_1^2, 1/(2b)) \\ x_3 &= \epsilon_3, \quad \epsilon_3 \sim \mathcal{N}(x_2^2, 1/(2c)). \end{aligned}$$

In this case, $\pi(\boldsymbol{x}) \propto e^{-f(\boldsymbol{x})}$ [8], where the Rosenbrock function $f(\boldsymbol{x}) : \mathbb{R}^3 \mapsto \mathbb{R}$ is

$$f(\boldsymbol{x}) = a(x_1 - \mu)^2 + b(x_2 - x_1)^2 + c(x_3 - x_2)^2.$$

Taking mixed partial derivatives of $\log \pi$ yields $\Omega_{13} = \Omega_{31} = 0$, i.e., $X_1 \perp\!\!\!\perp X_3 | X_2$.

3 NUMERICAL EXAMPLES

The numerical examples include a large-scale community climate model and a fire-detection satellite model. The climate model outputs satisfy an approximate Gaussian distribution, and the sparsity of the empirical precision matrix accelerates code verification and testing, but does not reveal obvious decoupling. On the other hand, the satellite outputs demonstrate some non-Gaussian behavior, and the conditional independence score Ω implies strong decoupling across subdomains.

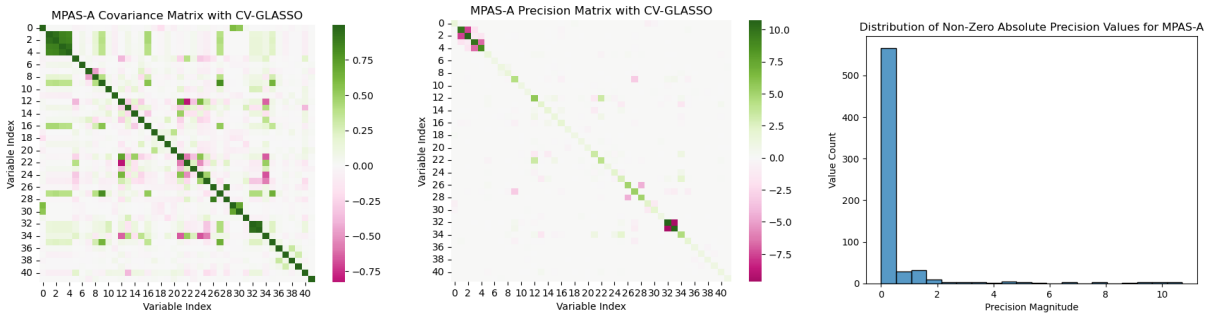
3.1 MPAS-A: A case study in code verification

For our first numerical example we consider outputs from a global atmospheric model, the Model Across Prediction Scales-Atmosphere (MPAS-A) [9]. While MPAS-A focuses solely on atmospheric processes of the climate, there is still substantial variety in the physics represented, ranging from evaporation to radiation mechanisms. We collect outputs from model runs with a quasi-uniform mesh that has 120 km resolution, 55 vertical levels and a total of of 40,962 cells. In its standard form, the MPAS-A model is deterministic. However, substantial stochastic variability exists due to the chaotic behavior of atmospheric systems.

We generate an ensemble of 430 model runs to represent the variability by applying machine level perturbations to the initial air temperature field and then running the model for 216 minutes (approximately 3.5 hours) of simulated time. Finally, each variable output is spatially averaged to a single value, following [10]. Including the default model outputs and excluding any constant or integer values results in 43 output variables. Due to an extremely high correlation between two of the remaining variables (> 0.9999), we omit one of them resulting in a final variable count of 42.

Due to the central limit theorem, the spatial averaging across all cells yields approximately Gaussian outputs. We verify this in three ways including (1) visual inspection of the marginal and 2D variable distributions across the ensemble of outputs; (2) inspection of QQ-Plots for all variables; and (3) application of the Shapiro-Wilk test for normality [11] with a cutoff of $p = 0.05$. All approaches indicated that, after spatial averaging, outputs of MPAS-A were approximately normal, such that the theory of section 2.1, specifically (3) suffices.

After standardizing the data, we calculate the covariance and precision matrices using Graphical Lasso, or GLASSO, an algorithm which uses an $l1$ regularization penalty to enforce sparsity in the computed precision matrix [12]. The strength of the $l1$ regularization term is scaled by the regularization parameter α, which is selected using k-fold cross validation, with $k = 5$. The cross validation results in $\alpha = 0.047$. The computed covariance and precision matrices can be seen in Figs. 1a and 1b, respectively. While approximately 38% of entries in the precision matrix are non-zero, most of these are small (as seen in Fig. 1c). This indicates that many of the edges in fact reflect weak conditional dependence; depending on the application, it may be beneficial to threshold the precision matrix even more aggressively.



(a) (b) (c)

Figure 1: MPAS-A results: (a) Covariance matrix, computed using cross-validated GLASSO; (b) The corresponding precision matrix. Approximately 38% of entries are non-zero; (c) Distribution of non-zero entries in the computed precision matrix for MPAS-A. Most non-zero entries are quite small, indicating weak conditional dependence.

From the precision matrix, Fig. 2 represents the probabilistic graph visually (with nodes and edges). The resulting graph is fully connected. Therefore we conclude this model is not suitable for decoupling. However, we can still take advantage of the structure of the graph. For instance, when refactoring code in atmospheric models (that is, making mathematically equivalent modifications), it is difficult to verify that errors have not been introduced. This is because even mathematically equivalent changes can result in substantially different outputs due to chaotic variability [13]. While more sophisticated methods of model verification certainly exist [10], a common initial check calculates the average or maximum difference between an accepted version of the model and a modified one, likely just for a few variables.

But if only a few variables are to be initially inspected, it would be helpful to know which ones are most likely to reflect unwanted changes in the code (i.e., the "canary" variables). Given the UGM above, we quantify highly interconnected variables with an approach known as Current Flow Betweenness Centrality (CFBC) [14]. Broadly, Betweenness Centrality calculates how likely a given node is to lie on the shortest path between two random nodes in the graph. CFBC interprets the edge weight as the capacity of current to flow between nodes, a fitting analogy for information flowing between conditionally dependent nodes in the graph. In Fig. 3a, we see the large variance in this CFBC measure, indicating some variables may be much better candidates for model verification tasks than others. Finally, Table 3b shows the top 3 variables, by betweenness score, along with their scientific descriptions.

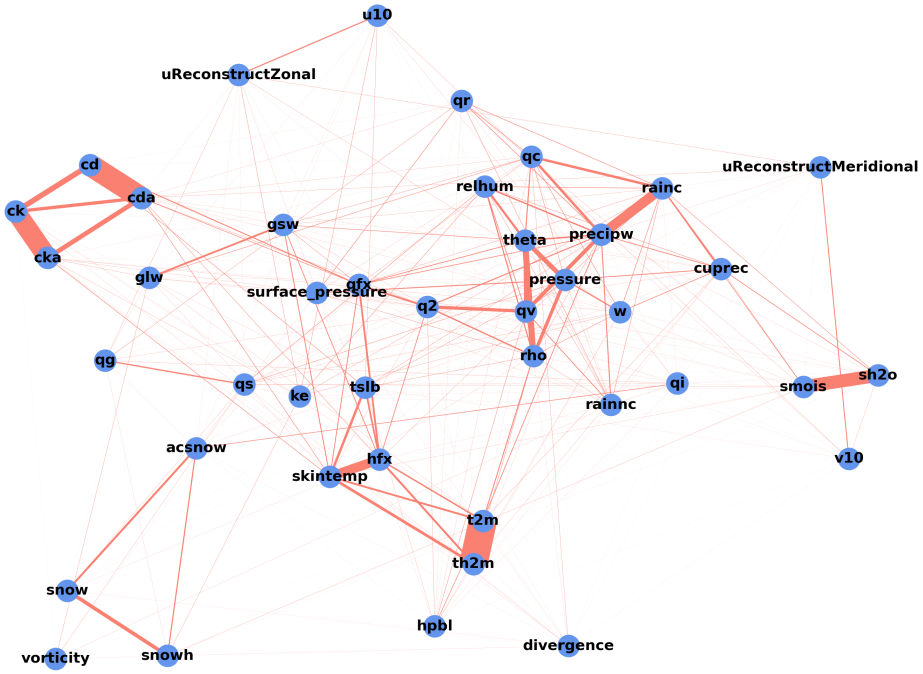
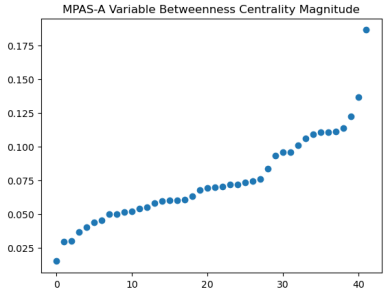


Figure 2: MPAS-A undirected probabilistic graphical model. Edge widths indicate magnitude of precision matrix entries. While this graph is fully connected, indicating that decoupling of model equations is unsuitable, substantial apparent structure enables more efficient approaches to tasks like code verification.



(a)

Variable	Betweenness Score	Description
skintemp	0.122212	Ground or water surface temperature
rainc	0.136409	Accumulated convective precipitation
precipw	0.186419	Precipitable water

(b)

Figure 3: (a) Distribution of CFBC betweenness scores for MPAS-A variables; (b) Top MPAS-A variables by betweenness score. These "canary" variables may be good candidates for initial verification of model outputs due to their centrality in the PGM.

3.2 Fire-detection satellite model: A case study of decoupling

The second application is a multidisciplinary model of an orbiting satellite which can detect and monitor forest fires in near real-time. The model comprises three major subdomains (orbit analysis, attitude control, and power analysis), contains both feedforward and feedback couplings, and outputs three QoIs (total torque, total power, and area of solar array). Subdomains refer to particular groupings of the coupled equations. Fig. 4 summarizes the fully coupled version, as described in [15]. That work used likelihood-based parameter estimation for the coupling variables without needing a fully coupled system analysis, while [16] developed adaptive surrogates to lesson the computational burden of a fixed-point iteration solver.

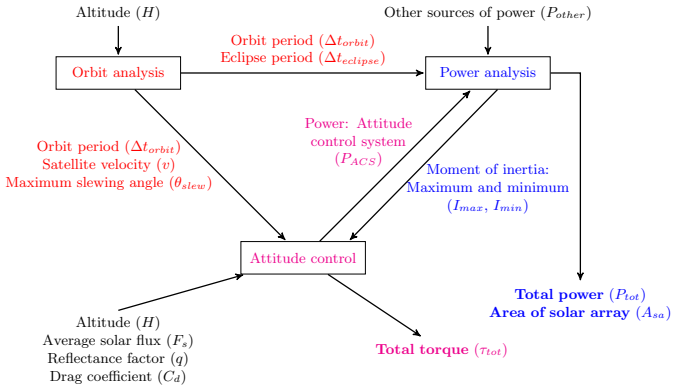
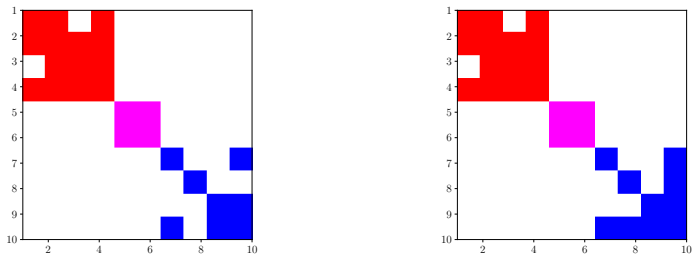


Figure 4: Schematic of the fully coupled fire detection satellite model. Red, magenta, and blue colors correspond to outputs of orbit, attitude, and power subdomains, respectively. The QoIs are bolded in the lower right corner.

To find the UGM of the satellite model, we assume data from a fully coupled model. We do not assume normality of the joint distribution *a priori*; i.e., the conditional independence score Ω is computed with the algorithm SING (Sparsity Identification in Non-Gaussian distributions) [17, 7], available through the software TRANSPORTMAPS [18].

Fig. 5 shows the adjacency matrix of the graph corresponding to the computed Ω for different values of β, a parameter that determines the richness of the transport map representation of the underlying distribution π. In particular, setting $\beta = 1$ restricts the map to be linear, which is appropriate when the data is multivariate normal. With $\beta > 1$, the transport map can describe non-Gaussian behavior. (More details about transport map representations and thresholding of the SING algorithm can be found in [7].) In this example, both $\beta = 1$ and 2 return disconnected components corresponding to the subdomains of the model, although the $\beta = 1$ graph is missing

an edge. Further testing showed convergence to the $\beta = 2$ graph, and we consider this result as the final UGM. Importantly, this graph implies that the model can be fully decoupled, as seen in Fig. 6.



(a) (b)

Figure 5: Learned adjacency matrices of the satellite model for (a) $\beta = 1$ (equivalent to assuming normality of the joint distribution; and (b) $\beta = 2$ (allowing for non-Gaussianity).

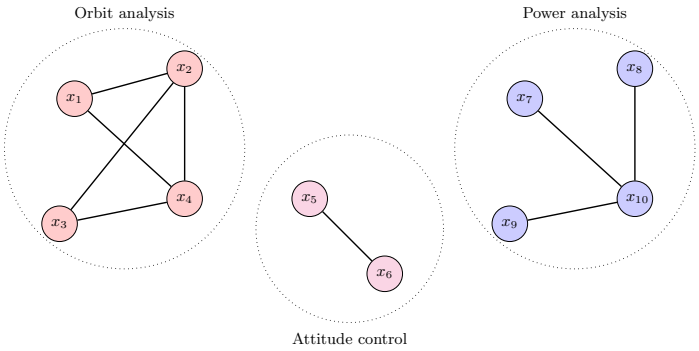


Figure 6: Fully decoupled satellite model, from computed conditional independence score Ω.

The identified model in Fig. 6 is consistent with the previous work in [19], which identified a fully decoupled model by comparing QoI distributions from candidate decoupled models with those of the fully coupled model. The procedure in [19] requires (1) distributions of the QoIs from the fully coupled model and all candidate models; and (2) computing the Kullback-Leibler divergence (or another metric on probability distributions) for all candidate decoupled models compared to the fully coupled model. In contrast, our approach only requires model outputs from the high-fidelity model and then computes the matrix of conditional independence scores Ω, which reveals the disconnected model components. In particular, QoIs can be reliably computed as outputs of a single subdomain, without even needing to run the others.

4 CONCLUSION

The complexity and computational cost of modern multi-physics codes is well known. It is apparent in the size of their code-bases and the possibly thousands of core-hours required to run them. In this work, we have shown that methods from the field of probabilistic graphical models can identify substantial structure in multi-physics models solely from output data. This structure can in turn identify important diagnostic variables for use in verification and correctness tasks and even possible model decouplings, thereby greatly reducing the complexity of these models.

Future work will continue to address the real-world utility of UGMs for computational models, for example, through realistic analysis of model errors (i.e., what is the root source of an error when code output changes?). Another intriguing avenue is to learn an UGM from experiments, rather than high-fidelity models, to design sparse multi-physics models from scratch.

5 Acknowledgements

The authors acknowledge Allison Baker and Michael Duda for their help and permission to use MPAS-A ensemble data from the PyCECT project [20]. TPB and RM acknowledge the Johnson & Johnson Foundation. RB acknowledges support from a Caltech instructorship, the AFOSR MURI on "Machine Learning and Physics-Based Modeling and Simulation" (FA9550-20-1-0358), and a DoD Vannevar Bush Faculty Fellowship (N00014-22-1-2790).

REFERENCES

[1] M. Kapteyn, J. Pretorius, and K. Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Computational Science*, 1(5):337–347, 2021.

[2] S. Xiao and T. Belytschko. A bridging domain method for coupling continua with molecular dynamics. *Computer methods in applied mechanics and engineering*, 193(17-20):1645–1669, 2004.

[3] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[4] S. L. Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[5] R. Morrison, R. Baptista, and E. Basor. Diagonal nonlinear transformations preserve structure in covariance and precision matrices. *Journal of Multivariate Analysis*, 190:104983, 2022.

[6] A. Spantini, D. Bigoni, and Y. Marzouk. Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, 19(1):2639–2709, 2018.

[7] R. Baptista, R. Morrison, O. Zahm, and Y. Marzouk. Learning non-Gaussian graphical models via hessian scores and triangular transport. *Journal of Machine Learning Research*, 25(85):1–46, 2024.

[8] F. Pagani, M. Wiegand, and S. Nadarajah. An n-dimensional rosenbrock distribution for MCMC testing. *arXiv preprint arXiv:1903.09556*, 2019.

[9] W. Skamarock, J. Klemp, M. Duda, L. Fowler, S.-H. Park, and T. Ringler. A Multiscale Nonhydrostatic Atmospheric Model Using Centroidal Voronoi Tesselations and C-Grid Staggering. *Monthly Weather Review*, 140(9):3090–3105, September 2012.

[10] D. Milroy, A. Baker, D. Hammerling, and E. Jessup. Nine time steps: ultra-fast statistical consistency testing of the Community Earth System Model (pyCECT v3.0). *Geoscientific Model Development*, 11(2):697–711, February 2018. Publisher: Copernicus GmbH.

[11] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, December 1965.

[12] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008.

[13] J. M. Rosinski and D. L. Williamson. The accumulation of rounding errors and port validation for global atmospheric models. *SIAM Journal on Scientific Computing*, 18(2), March 1997.

[14] U. Brandes and D. Fleischer. Centrality Measures Based on Current Flow. In V. Diekert and B. Durand, editors, *STACS 2005*, pages 533–544, Berlin, Heidelberg, 2005. Springer.

[15] S. Sankararaman and S. Mahadevan. Likelihood-based approach to multidisciplinary analysis under uncertainty. *Journal of Mechanical Design*, 134(3), 2012.

[16] A. Chaudhuri, R. Lam, and K. Willcox. Multifidelity uncertainty propagation via adaptive surrogates in coupled multidisciplinary systems. *AIAA Journal*, 56(1):235–249, 2018.

[17] R. Morrison, R. Baptista, and Y. Marzouk. Beyond normality: Learning sparse probabilistic graphical models in the non-Gaussian setting. In *Advances in Neural Information Processing Systems 30*, pages 2359–2369. Curran Associates, Inc., 2017.

[18] Transport Maps Team. TransportMaps v1.0. http://transportmaps.mit.edu.

[19] R. Baptista, Y. Marzouk, K. Willcox, and B. Peherstorfer. Optimal approximations of coupling in multidisciplinary models. *AIAA Journal*, 56(6):2412–2428, 2018.

[20] H. Xu, A. Baker, D. Milroy, and B. Johnson. NCAR/PyCECT. https://doi.org/10.5281/zenodo.6525719, May 2022.