VARIATIONAL DATA ASSIMILATION: 3D-VAR AND 4D-VAR

JAMES GABBARD

1. Introduction. Broadly, data assimilation algorithms fall into one of three categories: ensemble, variational, and hybrid. Ensemble methods like the Ensemble Kalman Filter rely on a large number of simultaneous simulations, and make use of statistics of the ensemble during the calculation. Variational methods operate sequentially, running a series of simulations and improving statistical estimates after each one. Hybrid methods use elements of both, as the name implies.

This work focuses on variational data assimilation, in particular the strong-constraint 4D-Var algorithm, which gained popularity in numerical weather predication and has been used by major weather centers in their forecasts [8]. Section 2 lays out the variational framework behind the 3D-Var and 4D-Var algorithms, while Section 3 showcases the application of 4D-Var to a data assimilation problem involving elastic rods. Section 4 brings the discussion to a close with with a brief overview of weak-constraint 4D-Var and the hybrid 4D-EnVar algorithm.

2. Formulation. The material here is synthesized from overviews by Bocquet, Ash, Nodat, and Courtier [4, 1, 5]. In the following, let the notation $\|\mathbf{x}\|_{\mathbf{A}}$ indicate the $\mathbf{A}$-norm of $\mathbf{x}$, so that $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^T\mathbf{A}\mathbf{x}$.

2.1. The Kalman update as a variational problem. Variational data assimilation begins with a re-examination of the update step of the Kalman update for systems with a linear observation function. Let $\mathbf{x}^t \in \mathbb{R}^n$ be a vector representing the true state of an n-dimenionsal system, and let $\mathbf{y}^t \in \mathbb{R}^m$ be a noiseless observation of the true state. If the observation function is linear, then $\mathbf{y}^t = \mathbf{H}\mathbf{x}^t$ for some observation matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$. In a data assimilation settings, neither the true state or the true observation is directly available to an analyst. However, we will assume that there is a background estimate of the state $\mathbf{x}^b$, constructed so that the background error error $\boldsymbol{\epsilon}^b = \mathbf{x} - \mathbf{x}^b$ is a mean zero random variable with covariance $\mathbf{B} = \mathrm{Cov}\big[\boldsymbol{\epsilon}^b\big]$. The system can also be observed by an imperfect instrument, which produces a noisy observation $\mathbf{y} = \mathbf{y}^t + \boldsymbol{\epsilon}^o$ with error $\boldsymbol{\epsilon}^o$ that has zero mean and a covariance matrix $\mathbf{R} \in \mathbb{R}^{m\times m}$.

The goal of statistical data assimilation is to construct an estimator $\mathbf{x}^a = f(\mathbf{x}^b, \mathbf{y})$ that more closely approximates the true state $\mathbf{x}^t$. The quality of the estimator can be quantified by properties of the error $\boldsymbol{\epsilon}^a = \mathbf{x}^a - \mathbf{x}^t$ and its covariance $\mathbf{P} = \mathrm{Cov}[\boldsymbol{\epsilon}^a]$. A common estimator is the Best Linear Unbiased Estimator (BLUE), which satisfies three properties:

- **Linear**: $\mathbf{x}^a$ is a linear function of $\mathbf{x}^b$ and $\mathbf{y}$.
- **Unbiased**: the error $\boldsymbol{\epsilon}^a$ is mean zero.
- **Best**: among all linear unbiased estimators, $\mathbf{x}^a$ minimizes $\mathrm{tr}(\mathbf{P})$.

Enforcing these three conditions leads to the well-known Kalman update

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^b), \tag{2.1}$$

where the Kalman gain $\mathbf{K}$ is an n by m matrix given by

$$\begin{align} \mathbf{K} &= \mathbf{B}\mathbf{H}^T\big(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\big)^{-1} \tag{2.2} \\ &= \big(\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\big)^{-1}\mathbf{H}^T\mathbf{R}^{-1}. \tag{2.3} \end{align}$$

The second form is related to the first via the Sherman-Morrison-Woodbury matrix identity. After the update, the posterior precision matrix $\mathbf{P}^{-1}$, is easily calculated from the precision of the background and the data:

$$\mathbf{P}^{-1} = \mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \tag{2.4}$$

A central idea of variational data assimilation is that the Best Linear Unbiased Estimate $\mathbf{x}^a$ is a also the minimizer of the cost function

$$J(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^b\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2_{\mathbf{R}^{-1}}. \tag{2.5}$$

To show the equivalence, note that critical points of $J(\mathbf{x})$ satisfy

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{x}} = \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^b) + \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{x} - \mathbf{y}) = \mathbf{0}, \tag{2.6}$$

which can be algebraically re-arranged to recover the Kalman update:

$$\mathbf{x}^a = \mathbf{x}^b + (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\big(\mathbf{y} - \mathbf{H}\mathbf{x}^b\big).$$

The Hessian matrix of $J(\mathbf{x})$ is the posterior precision,

$$\frac{\mathrm{d}^2 J}{\mathrm{d}\mathbf{x}^2} = \mathbf{P}^{-1} = \mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}. \tag{2.7}$$

Because $\mathbf{B}$ and $\mathbf{R}$ are positive definite, their inverses are as well, and so is $\mathbf{P}^{-1}$. Thus $J(\mathbf{x})$ is a quadratic function with a positive definite Hessian, indicating that is has only a single critical point at its global minimum.

The cost function 2.5 can also be motivated by maximum likelihood estimation. If the background error and observation error assumed to be independent zero-mean Gaussian random variables with covariance matrices $\mathbf{B}$ and $\mathbf{R}$ respectively, then the probability of obtaining a background state $\mathbf{x}_b$ and observations $\mathbf{y}$ from a true state $\mathbf{x}$ is given by

$$\mathcal{L}(\mathbf{x}) = p(\mathbf{x}_b, \mathbf{y}|\mathbf{x}) \sim \exp\left(\frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^b\right\|^2_{\mathbf{B}^{-1}}\right)\exp\left(\frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2_{\mathbf{R}^{-1}}\right). \tag{2.8}$$

Taking a logarithm, it becomes clear that $\log \mathcal{L}(\mathbf{x}) = -J(\mathbf{x}) + C$ for some constant C, so that minimizing J is equivalent to maximizing $\mathcal{L}$. This is in turn identical to performing a Bayesian estimation of $\mathbf{x}^t$, where $\mathbf{x}^t$ is given a Guassian prior with mean $\mathbf{x}_b$ and covariance $\mathbf{B}$, and the observation noise is assumed Gaussian with covariance $\mathbf{R}$. In this case the posterior for $\mathbf{x}^t$ is also a Gaussian distribution, so that the posterior mean and the MAP estimate of $\mathbf{x}^t$ coincide and both minimize $\mathbf{J}(\mathbf{x})$.

2.2. 3D-Var. Having rephrased the Kalman update as a minimization, the 3D-Var algorithm is simply the application of any iterative gradient-based optimization algorithm to minimize the cost function $J(\mathbf{x})$. For most choices, this means that the algorithm is entirely deterministic - the statistical or probabilistic side of the algorithm ends once the error and observation covariances $\mathbf{B}$ and $\mathbf{R}$ have been properly quantified. Unlike the standard Kalman update, the generalization of 3D-Var problems with a nonlinear observation function $\mathbf{y}^t = H(\mathbf{x})$ is clear and immediate, requiring only that the cost function be updated to

$$J(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^b\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - H(\mathbf{x})\|^2_{\mathbf{R}^{-1}}. \tag{2.9}$$

The nonlinearity prevents the posterior covariance from being written in closed form. However, a common work-around is to continue to use the Hessian of the cost function as as a first approximation. If the minimization is done with a Quasi-Newton algorithm, then an approximation of the Hessian of the cost function or its inverse is usually already available at convergence.

If the optimal state $\mathbf{x}^a$ is assumed to be close to the background state $\mathbf{x}_b$, then nonlinear observation function is typically linearized about the background state. This variant is known as incremental 3D-Var. For convenience, the object of inference is typically changed to the increment $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_b$. Defining a linearized observation operator $\mathbf{H} = H'(\mathbf{x}_b)$ and the innovation vector $\mathbf{d} = \mathbf{y} - H(\mathbf{x}_b)$, the resulting cost function can be written as

$$J(\mathbf{x}) = \frac{1}{2}\|\delta\mathbf{x}\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{H}\delta\mathbf{x} - \mathbf{d}\|^2_{\mathbf{R}^{-1}}. \tag{2.10}$$

This objective is quadratic in $\delta\mathbf{x}$, and convex. If increased accuracy is needed after the initial minimization, the nonlinear observation is re-linearized about the estimated state, and the resulting quadratic cost function is minimized again. This process repeats until convergence. If the minimization of the quadratic cost function is performed using a Krylov-subspace method, such as Conjugate Gradients, then this incremental algorithm is equivalent to minimizing the nonlinear cost function (2.9) using a Newton-Krylov method.

Finally, if there are far fewer observations than unknowns, the 3D-Var minimization can be significantly accelerated using a dual formulation. Starting from the Kalman update

$$\mathbf{x}^a - \mathbf{x}^b = \mathbf{B}\mathbf{H}^T\big(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\big)^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^b),$$

it is convenient to write $\mathbf{x}^a - \mathbf{x}^b = \mathbf{B}\mathbf{H}^T\mathbf{w}$, where $\mathbf{w}$ solves the system

$$(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T)\mathbf{w} = \mathbf{y} - \mathbf{H}\mathbf{x}^b \tag{2.11}$$

or equivalently minimizes the dual cost function

$$\tilde{J}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\Big(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\Big)\mathbf{w} - \mathbf{w}^T\big(\mathbf{y} - \mathbf{H}\mathbf{x}^b\big), \tag{2.12}$$

which has gradient

$$\frac{\mathrm{d}\tilde{J}}{\mathrm{d}\mathbf{w}} = \Big(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T\Big)\mathbf{w} - \big(\mathbf{y} - \mathbf{H}\mathbf{x}^b\big). \tag{2.13}$$

The vector $\mathbf{w}$ has the same size as the observation $\mathbf{y}$, which lives in a much lower dimension that the state vector $\mathbf{x}$. Additionally, the cost function now involves multiplication by the covariances $\mathbf{B}$ and $\mathbf{R}$ instead of their inverses. As before, if the the minimization is performed with a Quasi-Newton method then an approximation of this Hessian is available at convergence. This dual formulation is usually called a Physical-space Statistical Analysis System (PSAS). For a nonlinear observation function the 3D-Var PSAS algorithm can be applied using an incremental approach analogous to incremental 3D-Var.

2.3. The 4D-Var cost function.. The 3D-Var algorithm produces a single estimate of a state given a prior and noisy observations of that state. For time dependent systems, we can improve this estimate by incorporating the system dynamics into the cost function. This is the essence of 4D-Var, which has two main variants: strong constraint 4D Var for deterministic systems, and weak constraint 4D-Var for stochastic systems or systems with model uncertainty. Here we will focus on strong constraint 4D-Var, and leave the weak constraint formulation for section 4.1.

Consider a discrete-time dynamical system that takes state $\mathbf{x}^k$ at time t_k. The system dynamics are determined by functions $M_k(\mathbf{x})$, so that

$$\mathbf{x}_{k+1} = M_k(\mathbf{x}_k). \tag{2.14}$$

As in the 3D-Var formulation, assume that there is background knowledge of the initial condition $\mathbf{x}_0$, called $\mathbf{x}_0^b$, with error covariance $\mathbf{B}$. There are also noisy observations of the system at each point in time, so that $\mathbf{y}_k = H(\mathbf{x}_k) + \boldsymbol{\epsilon}_k^o$ with $\mathrm{Cov}[\boldsymbol{\epsilon}_k^o] = \mathbf{R}_k$. 4D-Var minimizes the cost function

$$J = \frac{1}{2}\left\|\mathbf{x}_0 - \mathbf{x}_0^b\right\|_{\mathbf{B}^{-1}}^2 + \frac{1}{2}\sum_k \left\|\mathbf{y}_k - H_k(\mathbf{x}_k)\right\|_{\mathbf{R}_k^{-1}}^2 \tag{2.15}$$

over all possible trajectories $\{\mathbf{x}_k\}_{k\geq 0}$. Practically, this optimization is carried out over the initial condition $\mathbf{x}_0$, since the remaining states $\{\mathbf{x}_k\}_{k\geq 1}$ can be written directly as a function of the initial condition:

$$\mathbf{x}_k = [M_{k-1} \circ M_{k-2} \circ \ldots \circ M_0]\,(\mathbf{x}_0). \tag{2.16}$$

For convenience, let $M_{0\to k} = M_{k-1} \circ M_{k-2} \circ \ldots \circ M_0$ be the function mapping $\mathbf{x}_0$ to $\mathbf{x}_k$. Let $\mathbf{M}_{0\to k} = M'_{0\to k}(\mathbf{x}_0)$ be a linearization of this function around the current guess for $\mathbf{x}_0$, and let $\mathbf{H}_k = H'(\mathbf{x}_k)$ be linearizations of the observation functions about the resulting trajectory $\{\mathbf{x}_k\}_{k\geq 0}$. The gradient of J is then,

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{x}_0} = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \sum_k \mathbf{M}_{0\to k}^T \mathbf{H}_k^T \mathbf{R}^{-1}(H(\mathbf{x}_k) - \mathbf{y}) \tag{2.17}$$

and the Hessian of J is

$$\frac{\mathrm{d}^2 J}{\mathrm{d}\mathbf{x}_0^2} = \mathbf{B}^{-1} + \sum_k \mathbf{M}_{0\to k}^T \mathbf{H}_k^T \mathbf{R}^{-1}\mathbf{H}_k \mathbf{M}_{0\to k}, \tag{2.18}$$

which serves as an estimate of the posterior precision for $\mathbf{x}_0$, here denoted $\mathbf{P}_0^{-1}$. Each term in the summation is at least positive semi-definite, so that the posterior precision increases as more and more data is observed.

Like 3D-Var, the 4D-Var cost function can be be derived from a maximum likelihood estimation of $\mathbf{x}_0^a$, where the background and observation errors are assumed to have a Guassian distribution. This is identical to a MAP estimate taken from a Bayesian model with a Gaussian prior for $\mathbf{x}_0$ and Gaussian observation noise. If the dynamics and the observation are linear, the optimal trajectory found by minimizng the 4D-Var cost function is also equivalent to the trajectory that would be estimated by a Kalman filter in the absence of model error.

As in 3D-Var, there is also a dual or PSAS formulations of 4D-Var. The dual vector $\mathbf{w}$ now must be the same size as the total observation vector $[\mathbf{y}_0,\, \mathbf{y}_1, \ldots,\, \mathbf{y}_k]$. This somewhat diminishes the usefulness of PSAS for time-dependent problems, since the the total number of observations can be much higher than the number of observations at a single instant. Consequently, the details of 4D-Var PSAS are omitted here and left to [5].

Finally, there is an incremental version of 4D-Var. Assuming that the optimal initial state $\mathbf{x}_0^a$ is not far from the background state $\mathbf{x}_0^b$, the dynamics $M_{0\to k}$ and observations H_k can be linearized about the background trajectory $\mathbf{x}_k^b$. The initial increment $\delta\mathbf{x}_0 = \mathbf{x}_0 - \mathbf{x}_0^b$ becomes the object of inference, and the later-time increments $\delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_k^b$ are related to the initial increment through the linearized dynamics $\delta\mathbf{x}_k = \mathbf{M}_{0\to k}\delta\mathbf{x}_0$. Introducing the innovation vectors $\mathbf{d}_k = H(\mathbf{x}_k^b) - \mathbf{y}_k$, the entire 4D-Var cost function can be rewritten as

$$J = \frac{1}{2}\|\delta\mathbf{x}_0\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_k \|\mathbf{H}_k\delta\mathbf{x}_k - \mathbf{d}_k\|^2_{\mathbf{R}_k^{-1}}. \tag{2.19}$$

which has gradient

$$\frac{\mathrm{d}J}{\mathrm{d}\delta\mathbf{x}_0} = \mathbf{B}^{-1}\delta\mathbf{x}_0 + \sum_k \mathbf{M}_{0\to k}^T\mathbf{H}_k^T\mathbf{R}^{-1}(\mathbf{H}_k\delta\mathbf{x}_k - \mathbf{d}_k) \tag{2.20}$$

and a Hessian equivalent to (2.18).

2.4. Linearization and Adjoints. The effectiveness of any of the 4D-Var algorithms discussed so far hinges on the evaluation of the linearized dynamics $\mathbf{M}_{0\to k}$, commonly called the tangent linear model, and their transpose $\mathbf{M}_{0\to k}^T$, commonly called the adjoint model. This section focuses on the linearization of each individual step $\mathbf{x}_{k+1} = M_k(\mathbf{x}_k)$, yielding matrices $\mathbf{M}_k$ that can be used to build the full tangent and adjoint operators:

$$\begin{aligned}\mathbf{M}_{0\to k} &= \mathbf{M}_{k-1}\cdot\mathbf{M}_{k-2}\cdot\ldots\cdot\mathbf{M}_1\cdot\mathbf{M}_0,\\ \mathbf{M}_{0\to k}^T &= \mathbf{M}_0^T\cdot\mathbf{M}_1^T\cdot\ldots\cdot\mathbf{M}_{k-2}^T\cdot\mathbf{M}_{k-1}^T.\end{aligned}$$

One strategy for producing tangent and adjoint dynamics can begins directly with a continuous time model. Here we assume that the states $\mathbf{x}_k$ at times t_k are samples of a dynamical system defined by the ODE $\dot{\mathbf{x}} = \mathbf{f}(t,\mathbf{x})$. To reflect the fact that a state $\mathbf{x}(t)$ is entirely determined by the initial condition $\mathbf{x}_0 = \mathbf{x}(0)$, we will write $\mathbf{x} = \mathbf{x}(t,\mathbf{x}_0)$. This two argument function is then constrained by the PDE

$$\frac{\partial\mathbf{x}(t,\mathbf{x}_0)}{\partial t} = \mathbf{f}(t,\mathbf{x}(t,\mathbf{x}_0)). \tag{2.21}$$

Differentiating with respect to the initial condition and swapping the order of differentiation on the right hand side then produces an ODE for the time evolution of the sensitivity $\partial\mathbf{x}(t,\mathbf{x}_0)/\partial\mathbf{x}_0$:

$$\frac{\partial}{\partial t}\left[\frac{\partial\mathbf{x}(t,\mathbf{x}_0)}{\partial\mathbf{x}_0}\right] = \frac{\partial\mathbf{f}(t,\mathbf{x}(t,\mathbf{x}_0)}{\partial\mathbf{x}(t,\mathbf{x}_0)}\left[\frac{\partial\mathbf{x}(t,\mathbf{x}_0)}{\partial\mathbf{x}_0}\right] \tag{2.22}$$

This is a linear ODE, commonly called the tangent linear model (TLM) or the forward sensitivity equations. Using the initial condition $\mathrm{d}\mathbf{x}_0/\mathrm{d}\mathbf{x}_0 = \mathbf{I}$, this ODE can be integrated forward in time to produce the matrix $\mathbf{M}_{0\to k} = \mathrm{d}\mathbf{x}(t_k)/\mathrm{d}\mathbf{x}_0$. More practically, given the trajectory $\mathbf{x}(t)$ about which the dynamics are linearized, the matrix-vector product $\mathbf{M}_{0\to k}\mathbf{y}$ can be evaluated by solving the initial value problem

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} &= \left.\frac{\partial\mathbf{f}}{\partial\mathbf{x}}\right|_{\mathbf{x}(t)}\boldsymbol{\lambda}(t),\\ \boldsymbol{\lambda}(t_0) &= \mathbf{y}\end{aligned}$$

and recording the final state $\boldsymbol{\lambda}(t_k) = \mathbf{M}_{0\to k}\mathbf{y}$. The adjoint dynamics $\mathbf{M}_{0\to k}^T$ can be evaluated similarly, using a linear system that depends on the transposed linear dynamics $\partial\mathbf{f}/\partial\mathbf{x}^T$. To evaluate the matrix-vector product $\mathbf{M}_{0\to k}^T\mathbf{y}$, the time integration is conducted *backwards* in time from t_k to t_0, using $\mathbf{y}$ as a final condition:

$$\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} = \left.\frac{\partial f}{\partial x}\right|_{x(t)}^T \boldsymbol{\lambda}(t),$$

$$\boldsymbol{\lambda}(t_k) = \mathbf{y}.$$

This produces an initial value $\lambda(t_0) = \mathbf{M}_{0\to k}^T\mathbf{y}$.

An alternative strategy for tangent and adjoint computation is to first discretize the continuous system, and then linearize the discrete model. Of particular usefulness is the linearization of dynamics in which the update rule $\mathbf{x}_{k+1} = M(\mathbf{x}_k)$ is determined implicitly through a functional relation $G(\mathbf{x}_k, \mathbf{x}_{k+1}) = 0$. A common example of this is an implicit Euler discretization of the continuous system $\dot{x} = f(x,t)$, in which each time steps obey the relation

$$G(\mathbf{x}_k, \mathbf{x}_{k+1}) = \mathbf{x}_{k+1} - \mathbf{x}_k - \Delta t f(\mathbf{x}_{k+1}, t_{k+1}) = 0. \tag{2.23}$$

Differentiating $G(\mathbf{x}_k, \mathbf{x}_{k+1})$ with respect to $\mathbf{x}_k$ gives the relation

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}_k} G(\mathbf{x}_k, M(\mathbf{x}_k)) = 0 \quad \rightarrow \quad \frac{\partial G}{\partial \mathbf{x}_k} + \frac{\partial G}{\partial \mathbf{x}_{k+1}}\mathbf{M}_k = 0,$$

which can be solved to give either the tangent or adjoint dynamics:

$$\mathbf{M}_k = -\left[\frac{\partial G}{\partial \mathbf{x}_{k+1}}\right]^{-1} \frac{\partial G}{\partial \mathbf{x}_k} \tag{2.24}$$

$$\mathbf{M}_k^T = -\frac{\partial G}{\partial \mathbf{x}_k}^T \left[\frac{\partial G}{\partial \mathbf{x}_{k+1}}\right]^{-T}. \tag{2.25}$$

Here all derivatives are evaluated at the solution point $(\mathbf{x}_{k+1},\, \mathbf{x}_k)$. Practically, if the forward dynamics are calculated by solving $G(\mathbf{x}_k, \mathbf{x}_{k+1}) = 0$ for $\mathbf{x}_{k+1}$ using Newton's method, then the Jacobian matrix $\partial G/\partial \mathbf{x}_{k+1}$ is already available for use in the tangent or adjoint passes.

2.5. The complete 4D-Var algorithm. Like 3D-Var, the 4D-Var algorithm is simply the application of any gradient-based minimization algorithm to either the full or incremental 4D-Var cost function. In the fully nonlinear case, this requires the evaluation of the gradient

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{x}_0} = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \sum_k \mathbf{M}_{0\to k}^T \mathbf{H}_k^T \mathbf{R}^{-1}(H(\mathbf{x}_k) - \mathbf{y}), \tag{2.26}$$

and in the incremental case this is replaced by the incremental gradient (2.20). Here we focus on the fully nonlinear form, and note that the incremental form can recovered simply be replacing the dynamics $M_k(\cdot)$ and observations $H_k(\cdot)$ with their linearized counterparts.

The evaluation of the gradient begins with a forward pass, in which the system dynamics are integrated to produce a trajectory $\{\mathbf{x}_k\}_{k\geq 0}$. At each step, we also record the intermediate quantities

$$\Delta_k = \mathbf{H}_k^T \mathbf{R}^{-1}(H(\mathbf{x}_k) - \mathbf{y}), \tag{2.27}$$

so that second term in the gradient can be written succinctly as $\sum_k \mathbf{M}_{0\to k}^T \Delta_k$. The key to efficiently evaluating this sum is the factorization

$$\begin{aligned}\sum_k \mathbf{M}_{0\to k}^T \Delta_k &= \sum_k \mathbf{M}_0^T \cdot \mathbf{M}_1^T \cdot \ldots \cdot \mathbf{M}_{k-1}^T \Delta_k \\ &= \Delta_0 + \mathbf{M}_0^T(\Delta_1 + \mathbf{M}_1^T(\Delta_2 + \ldots + \mathbf{M}_{k-2}^T(\Delta_{k-1} + \mathbf{M}_{k-1}^T \Delta_k))).\end{aligned}$$

This is reminiscent of the Horner factorization of a polynomial, and consistent with the backwards-in-time adjoint system described in the previous section. The calculation begins Δ_k and proceeds backwards to Δ_0, at each step multiplying by $\mathbf{M}_j$ and adding the next Δ_j. Once this adjoint pass is complete, the addition of the prior term $\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b)$ completes the gradient evaluation.

Each minimization step requires one forward integration of the model and one backwards (adjoint) pass to calculate the gradient. Consequently, the computational cost of the algorithm depends mainly on three factors: the complexity of the forward system, the complexity of the adjoint system, and the number of iterations needed for convergence. If the forwards pass is done with an explicit time integration, then the adjoint pass is typically at least as expensive as the forward pass. If the forwards pass proceeds by iteratively solving a nonlinear system at each step, the adjoint pass may be considerably faster due to its linear, non-iterative nature. In either the case, the minimization algorithm is a key component, and typically a Quasi-Newton method is chosen to accelerate convergence without explicitly computing the exact Hessian matrices [5].

3. Application: Elastic Rods. Strong constraint 4D-Var is most commonly discussed in the context of numerical weather prediction. However, it is broadly applicable to any deterministic dynamical system, including the dynamics of elastic rods. This example application retains the "4D" aspect of the algorithm by considering a 3D time-dependent system, but requires vastly less computational power than a full 3D nonlinear PDE.

The elastic rod model used here is geometrically nonlinear, but not elastically nonlinear. This means that while the rod itself undergoes large deformations, and any individual material element within the rod is deformed only slightly. A very brief description is provided in this section, while a more thorough treatment can be found in the Appendix and in references [3, 2, 10]. The centerline of a discrete rod is represented by a set of connected vertices $\{\mathbf{x}_i\}$ moving with velocities $\{vbv_i\}$, as well as set of angular degrees of freedom θ^i that describe the orientation of each rod's cross section on each connecting edge. The elastic energy stored in such a rod can be broken into components related to stretching of the centerline, bending of the centerline, and twisting of the cross section about the centerline. The total elastic energy E of the rod is then the sum of these components. The elastic forces $\mathbf{F}_i$ acting at each vertex are expressed as the derivative of the elastic energy with respect to a centerline perturbation,

$$\mathbf{F}_i = \frac{\partial E}{\partial \mathbf{x}_i}. \tag{3.1}$$

Similarly, the twisting moment on each edge is expressed as the derivative of elastic energy with respect to the angular parameter $\mathbf{theta^i}$, so that

$$T^i = \frac{\partial E}{\partial \theta^i}. \tag{3.2}$$

The mass associated with each vertex on the rod is taken to be $M_i = \rho A \bar{l}_i$, where A and l_i are the cross sectional area and length of a single edge and ρ is the density of the material. Similarly, the moment of inertia of each edge segment is taken to be $I^i = \rho \bar{l}^i J$, where J is the polar moment of inertia of the cross section. Rod dynamics are then described by the second order system

$$\begin{aligned} M_i \ddot{\mathbf{x}}_i &= \mathbf{F}_i + \mathbf{F}_i^{\text{ext}}, \\ I^i \ddot{\theta}^i &= T_i + T_{\text{ext}}^i. \end{aligned}$$

As described in [3], twisting dynamics typically have a much smaller time scale than centerline dynamics. Thus for thin rods, its convenient to take $I^i = 0$, indicating that the rod's twisting degrees of freedom are always in a quasi-static equilibrium. With this assumption, the system can be rephrased as a first order differential-algebraic equation

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{V} \\ \mathbf{M}\dot{\mathbf{V}} &= \mathbf{F}(\mathbf{X}, \boldsymbol{\theta}) + \mathbf{F}_{\text{ext}}, \\ 0 &= T(\mathbf{X}, \boldsymbol{\theta}) + T_{\text{ext}}, \end{aligned}$$

Where $\mathbf{X}$, $\mathbf{V}$, $\boldsymbol{\theta}$ are vectors that group together all position, velocity, rotation degrees of freedom. If this DAE is integrated in time with the Backwards Difference Method, each time step is determined from the last by solving the nonlinear system

$$\begin{aligned} \mathbf{X}_{k+1} - \mathbf{X}_k - \Delta t \mathbf{V}_{k+1} &= 0 \\ \mathbf{M}\mathbf{V}_{k+1} - \mathbf{M}\mathbf{V}_k - \Delta t F(\mathbf{X}_{k+1}, \boldsymbol{\theta}_{k+1}) &= 0 \\ -\Delta t T(\boldsymbol{X}_{k+1}, \boldsymbol{\theta}_{k+1}) &= 0 \end{aligned}$$

Collecting all degrees of freedom into a vector $q = [\mathbf{X}, \mathbf{V}, \boldsymbol{\theta}]$, this update has the form $G(q_k, q_{k+1}) = 0$, with partial derivatives

$$\frac{\partial G}{\partial q_k} = \begin{bmatrix} -Id & 0 & 0 \\ 0 & -M & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial G}{\partial q_{k+1}} = \begin{bmatrix} Id & -\Delta t Id & 0 \\ -\Delta t F_x & M & -\Delta t F_\theta \\ -\Delta t T_x & 0 & -\Delta t T_\theta \end{bmatrix}.$$

As derived in Section 2.4, these partial derivatives can be used to construct the adjoint and tangent dynamics for each step of the time integration.

3.1. Problem Specification. The estimation problem considered here is to infer the correct trajectory of a rod's centerline, given by the positions $\mathbf{x}(s,t)$ and velocities $\mathbf{v}(s,t)$. The inference is based on noisy observations of the centerline at selected points s_i along the length and selected times $t_j \in [t_0, t_f]$, so that the entire dataset is $\mathbf{y}_{ij} = \mathbf{x}(s_i, t_j) + \boldsymbol{\epsilon}_{ij}^o$. Here the selected points s_i are at $s = \{L/4,\ L/2,\ 3L/4,\ L\}$, where L is the length of the rod, and the observations are taken at equally spaced points between t_0 and t_f. There is also a provided background state for the initial positions $\mathbf{x}_0^b(s)$ and velocities $\mathbf{v}_0^b(s)$. The covariances of $\boldsymbol{\epsilon}_{ij}^o$ and the background states are assumed to be known.

The specific trajectory studied in this section is a rod hanging vertically from a clamped top end. The rod is subject to the force of gravity, acting in the negative z direction of an $(x,\ y,\ z)$ coordinate system. The clamped end is forced to trace out elliptical path in the $(x,\ y)$ plane, with radii $(r_x,\ r_y)$ and angular frequency ω; the rest of the rod swings freely below, as illustrated in Figure 1. To produce observations and a prior for the initial condition, a reference trajectory is generated starting from a resting state, and after suitable time has passed for the dynamics to become interesting, the reference trajectory is sampled at 11 equally spaced points. The first of these is taken as the initial condition, and the background states are obtained by adding a random perturbation to this state. The perturbations in each component of $\mathbf{x}_0^b(s)$ and $\mathbf{v}_0^b(s)$ are independent, and drawn from a Gaussian process with mean zero and covariance

$$\begin{aligned} B_x(s_1, s_2) &= \sigma_{xb}^2 \exp\left(-\frac{|s_1 - s_2|^2}{2L_{xb}^2}\right), \\ B_v(s_1, 2_2) &= \sigma_{vb}^2 \exp\left(-\frac{|s_1 - s_2|^2}{2L_{vb}^2}\right). \end{aligned}$$

The observations $\mathbf{y}_{ij}$ are generated by adding noise to the positions observed in the other ten sampled states. The noise values in each component of each position are assumed assumed to be independent mean-zero Gaussian random variables, each with standard deviation σ_y.

The true covariances $B_x(s_1, s_2)$ and $B_v(s_1, 2_2)$ are ill-suited for use in a 4D-Var algorithm: because both are smooth kernels, their eigenvalues decay rapidly, leading to a prior covariance matrix $\mathbf{B}$ which is effectively singular. This can be remedied by localization, which reduces the covariance between points with a large separation $|s_1 - s_2|$ and reduces the condition number of the covariance matrix. The localized background covariance function used here is

$$\tilde{B}_x(s_1, s_2) = (1 - \max(0, |s_1 - s_2|/L_{\text{loc}}))B_x(s1, s_2),$$

and similarly for $\tilde{B}_v(s_1, s_2)$. Here we take $L_{\text{Loc}} = L$, for simplicity. These localized matrices are used to generate the background covariance $\mathbf{B}$ used in all of the following optimizations.

3.2. Implementation Notes.

The implementation and testing of the adjoint dynamics for Discrete Elastic Rods was nontrivial, to say the least. However, because the rod system described here is small (roughly 170 unknowns), its feasible to calculate and store all elements of the tangent matrices $\mathbf{M}_k$ during the time integration, as well as to calculate $\mathbf{B}^{-1}$ and $\mathbf{C}^{-1}$ explicitly. This allows the adjoint pass to be implemented by directly calculating the matrix-vector products described in Section 2.5, and it also allows for the direct calculation of the Hessian matrix $\mathrm{d}^2 J/\mathrm{d}\mathbf{x}_0^2$. The minimization is done both with the ADAM optimizer [9], which does not require as Hessian, and with Newton's method, which does make use of the full Hessian. All presented results use a rod that is discretized with 25 segments, and integrated in time over 100 steps covering two full cycles of the elliptical forcing.

3.3. Results.

To investigate the effectiveness of 4D-Var on the rod assimilation problem, the algorithm is run on test cases with different characteristic scales for the background and observation covariances. Table 3.3 lists the covariance parameters for all 4 test cases, which are expressed in terms of a characteristic length L and characteristic velocity $\mathbf{v}$ which is the average of the norm of the initial velocity at each vertex.

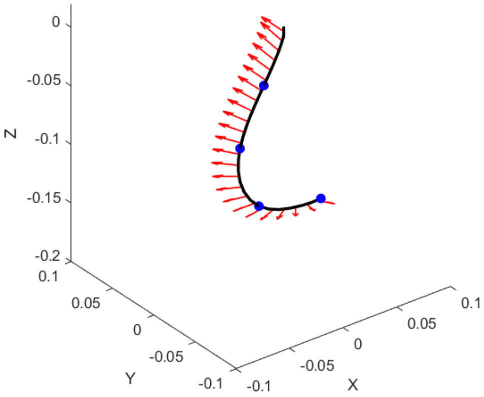


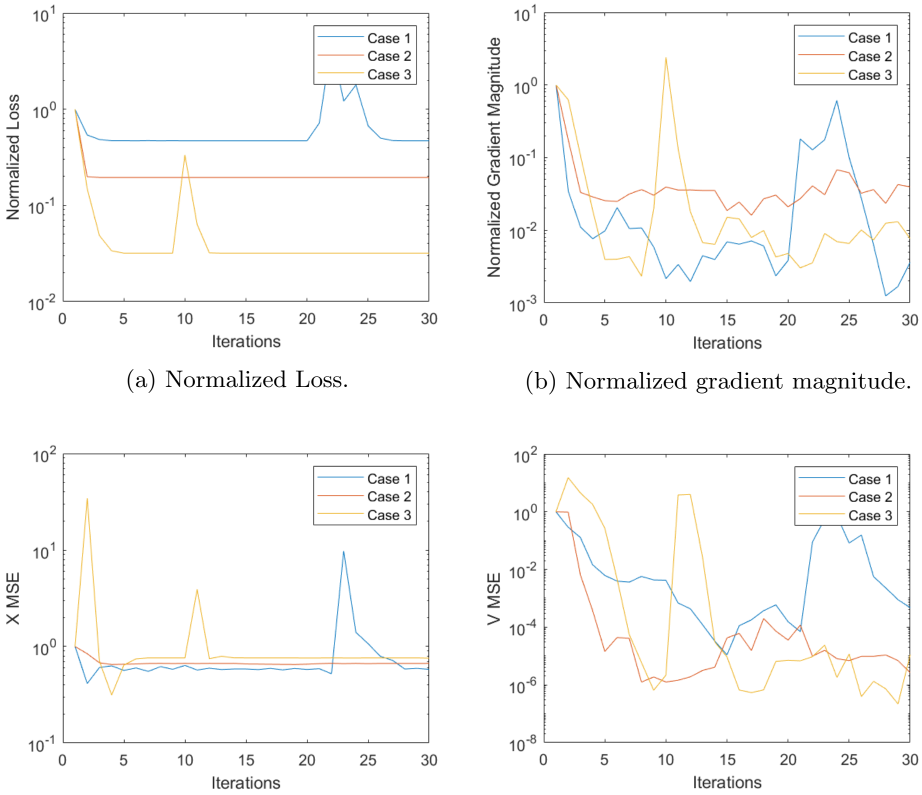
Fig. 1: A hanging rod with oscillates due to an elliptical drive at its fixed upper end. Red arrows indicate centerline orientation, while blue dots indicate the observation points on the centerline.

Case	Description	σ_{xb}/L	$\sigma_{vb}/\bar{v}$	σ_y/L
1	Large **B**, Large **R**	0.05	2.0	0.10
2	Small **B**, Small **R**	0.01	0.4	0.015
3	Small **B**, Large **R**	0.01	0.4	0.05
4	Large **B**, Small **R**	0.05	2.0	0.015

For Cases 1 to 3, Newton's method performed well, while ADAM failed to reach any acceptable converged state. This mismatch may be due to the differing stretching and bending stiffnesses in the rod, which could lead to an objective with a Hessian that is strongly "elliptical". This is difficult for purely-gradient-based algorithms, but can be compensated for in algorithms that use an exact or approximate Hessian. The 4th case led to divergence for both algorithms, indicating that 4D-Var is most difficult when the precision of the observations far exceeds the precision of the background state.

Figure 2 shows the convergence history of Newton's method for cases 1 to 3, including the cost function, the norm of the gradient, and the mean-squared error in the position and velocity variables when compared to the reference trajectory. All of these quantities are normalized by their initial values, for a fair comparison. Surprisingly, the initial velocities are assimilated much faster and more effectively than the initial positions, which undergo very little change during the calculation. In all cases the loss and gradient plateaus after about a dozen iterations.

Figure 3 examines the prior and posterior covariance matrices for the z component of the centerline velocity, using data from Case 3. There is a noticeable decrease in the variance of the velocity midway down the centerline, which is the region that experiences the largest displacements during the rod's oscillating trajectory. The covariances of position, not shown, are virtually unchanged, consistent with the earlier finding that velocities are much more easily estimated than positions for this model.



(a) Normalized Loss.

(b) Normalized gradient magnitude.

(c) MSE in the initial positions.

(d) MSE in the initial velocities.

Fig. 2: Convergence history of cases 1 to 3, minimized with Newton's method.

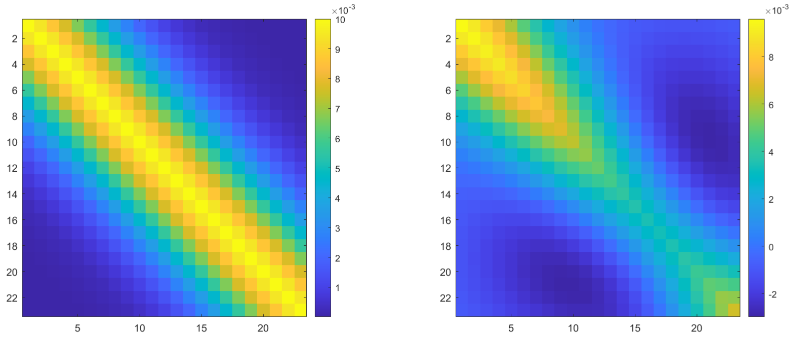


Fig. 3: Prior (left) and posterior (right) covariance of the x velocity components for Case 3.

Finally, Figure 4 displays the components of the posterior precision directly attributable to each set of observations, $\mathbf{M}_{0\to k}^T \mathbf{H}_k^T \mathbf{R}^{-1} \mathbf{H}_k \mathbf{M}_{0\to k}$ for observations at the second, fifth, and tenth observation points. The displayed components are for the x direction positions and velocities. While the x position components are directly observed, and thus affected strongly by specific values of the observation errors, the velocity precisions are considerably more diffuse, since they are assimilated indirectly through the dynamics. The precision values also decrease at later times, indicating that observations further away from the initial condition are less and less useful in estimating it.

4. Conclusions.

As demonstrated in the previous section, 4D-Var is an effective data-assimilation method for nonlinear dynamical systems. Here we conclude with a brief discussion of two variants of 4D-Var: weak constraint 4D-Var, and hybrid methods like 4D-EnVar, the latter of which is cutting edge in numerical weather prediction even today.

4.1. Weak Constraint 4D-Var.

Systems with model uncertainty or stochastic dynamics are addressed by weak-constraint 4D-Var. In this formulation, the dynamical system is of the form $\mathbf{x}_k = M(\mathbf{x}_k) + \boldsymbol{\epsilon}^d$, where the noise in the dynamics $\boldsymbol{\epsilon}^d$ is mean-zero with covariance $\mathbf{Q}_k$. The corresponding cost function on a trajectory $\{\mathbf{x}_k\}$ is given by

$$J = \frac{1}{2}\left\|\mathbf{x}_0 - \mathbf{x}_0^b\right\|_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_k \|\mathbf{y}_k - H_k(\mathbf{x}_k)\|_{\mathbf{R}_k^{-1}} + \frac{1}{2}\sum_k \|\mathbf{x}_{k+1} - M_k(\mathbf{x}_k)\|_{\mathbf{Q}_k^{-1}}. \tag{4.1}$$

Because the dynamics are not deterministic, the initial condition $\mathbf{x}_0$ no longer determines the entire trajectory. Consequently, it is necessary to optimize over all of the states $\mathbf{x}_k$, dramatically increasing the dimension of the minimization. This increased dimension is a major drawback, and as a result the weak-constraint formulation was considered computationally intractable in numerical weather predication until relatively recently [6].

4.2. 4D-EnVar and Hybrid Assimilation.

In addition to the minimization algorithms discussed extensively in Section 2, many forecast tasks require the propagation of the background error covariance forward in time to the beginning of the next forecast window. This can be done by repeated evaluations of the tangent linear model, but this form of propagation does not take into account nonlinearities in the dynamics. In 4D-EnVar, an entire ensemble of trajectories beginning from slightly perturbed background states is used in place of the tangent linear model to propagate covariances forward in time [7]. This approach combines some of the best attributes of ensemble and variational assimilation, and can lead to increased accuracy in nonlinear systems with large background covariances.

Acknowledgments. Thank you all for a great semester of 16.940!

REFERENCES

[1] M. Asch, M. Bocquet, and M. Nodet, *Data assimilation: methods, algorithms, and applications*, SIAM, 2016.

[2] M. Bergou, B. Audoly, E. Vouga, M. Wardetzky, and E. Grinspun, *Discrete viscous threads*, ACM Transactions on Graphics (TOG), 29 (2010), pp. 1–10.

[3] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, *Discrete elastic rods*, in ACM SIGGRAPH 2008 papers, 2008, pp. 1–12.

[4] M. Bocquet, *Introduction to the principles and methods of data assimilation in geosciences*, Notes de cours, École des Ponts ParisTech, (2014).

[5] P. Courtier, *Dual formulation of four-dimensional variational assimilation*, Quarterly Journal of the Royal Meteorological Society, 123 (1997), pp. 2449–2461.

[6] P. Courtier, J.-N. Thépaut, and A. Hollingsworth, *A strategy for operational implementation of 4d-var, using an incremental approach*, Quarterly Journal of the Royal Meteorological Society, 120 (1994), pp. 1367–1387.

[7] D. Fairbairn, S. R. Pring, A. C. Lorenc, and I. Roulstone, *A comparison of 4dvar with ensemble data assimilation methods*, Quarterly Journal of the Royal Meteorological Society, 140 (2014), pp. 281–294.

[8] P. Gauthier, M. Tanguay, S. Laroche, S. Pellerin, and J. Morneau, *Extension of 3dvar to 4dvar: Implementation of 4dvar at the meteorological service of canada*, Monthly weather review, 135 (2007), pp. 2339–2354.

[9] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014).

[10] C. Lestringant, B. Audoly, and D. M. Kochmann, *A discrete, geometrically exact method for simulating nonlinear, elastic and inelastic beams*, Computer Methods in Applied Mechanics and Engineering, 361 (2020), p. 112741.

5. Appendix: Rod Modeling.

5.1. Continuous rods. The position of a continuous elastic rod is described by a 3D curve with an arc-length parametrization $\mathbf{x}(s)$, so that the tangent vector to the rod's centerline is simply $\mathbf{t}(s) = \mathbf{x}'(s)$. We assume that at each point the rod's cross section remains perpendicular to its centerline, and that the shape of the cross section does not deform. However, the rod is free to rotate about its centerline, and it is necessary to specify a unit vector $\mathbf{d}_1(s)$ perpendicular to $\mathbf{t}(s)$ which indicates the orientation of the cross section. Together the unit vectors $\mathbf{t}$, $\mathbf{d}_1$, and $\mathbf{d}_2 = \mathbf{t} \times \mathbf{d}$ form an orthonormal triad at each point on the curve, usually called an adapted frame.

Although $\mathbf{x}(s)$ and $\mathbf{d}_1(s)$ completely specify the kinematics of the rod, they are not sufficient to define its elastic energy. To do so it is necessary to specify a resting configuration for the rod, $(\bar{\mathbf{x}}(s), \bar{\mathbf{d}}_1(s))$, in which no elastic energy is stored. Once this is done, the elastic energy is defined in terms of the stretching, bending, and twisting strains of the rod. The stretching of an infinitesimal rod segment with length $\mathrm{d}l$ is defined in relation to its resting length $\mathrm{d}\bar{l}$, leading to a stretching strain $\epsilon = \mathrm{d}l/\mathrm{d}\bar{l} - 1$. The bending of a rod is quantified by the curvature normal vector, $\boldsymbol{\kappa}(s) = \mathbf{t}'(s) = \mathbf{x}''(s)$. This vector is perpendicular to the centerline, and so can be specified by two scalar components $\kappa_1(s) = \kappa(s) \cdot \mathbf{d}_1(s)$ and $\kappa_2(s) = \boldsymbol{\kappa}(s) \cdot \mathbf{d}_2(s)$. Lastly, the twist of a rod is determined by the rate at which the cross section rotates about the centerline, $\tau(s) = \mathbf{d}_1'(s) \cdot \mathbf{d}_2$. These definitions apply equally to the resting state, which has resting curvatures $\bar{\kappa}_1$, $\bar{\kappa}_2$ and resting twist $\bar{\tau}$.

For simplicity, assume that the material in the rod has a constant elastic modulus E and shear modulus G. Assume also that the cross sectional area A, area moment of inertia $\mathbf{I}$ (a tensor expressed in the $\mathbf{d}_1$, $\mathbf{d}_2$ frame), and polar moment of inertia J do not vary along the length of the rod. The elastic energy due to each form of strain is then

$$\begin{aligned} E_\epsilon &= \frac{EA}{2} \int \epsilon^2 \, \mathrm{d}s \,, \\ E_\kappa &= \frac{E}{2} \int \|\boldsymbol{\kappa} - \bar{\boldsymbol{\kappa}}\|_{\mathbf{I}} \, \mathrm{d}s \,, \\ E_\tau &= \frac{GJ}{2} \int (\tau - \bar{\tau})^2 \, \mathrm{d}s \,, \end{aligned}$$

and the total elastic energy is the sum $E_{\mathrm{el}} = E_\kappa + E_\tau + E_\epsilon$.

5.2. Discretization. In “Discrete Elastic Rods” [3] and it’s successor “Discrete Viscous Threads” [2], Bergou et al. develop a discrete analogue to the continuous theory described above. In this model, the centerline of the elastic rod is represented by a collection of vertices $\{\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_n\}$, connected in sequence by edge vectors $\mathbf{e}^i = \mathbf{x}_{i+1} - \mathbf{x}_i$ with lengths $l^i = \|\mathbf{e}^i\|$. The tangent vectors $\mathbf{t}^i = \mathbf{e}^i/l^i$ take the role of the continuous tangent $\mathbf{t}(s)$, and each edge carries additional directors $\mathbf{d}_1^i$ and $\mathbf{d}_2^i$ that form an orthonormal triad with $\mathbf{t}^i$.

The discrete stretching, bending, and twisting, strains in this model are adapted to the discrete description of the ceneterline and directors. the discrete stretch is tracked by the edge lengths l^i and resting edge lengths $\bar{l}^i$. The discrete curvature associated with each interior vertex is quantified by a curvature binormal vector

$$\boldsymbol{\kappa}_i = \frac{2\mathbf{t}^i \times \mathbf{t}^{i-1}}{1 + \mathbf{t}^i \cdot \mathbf{t}^{i-1}}. \tag{5.1}$$

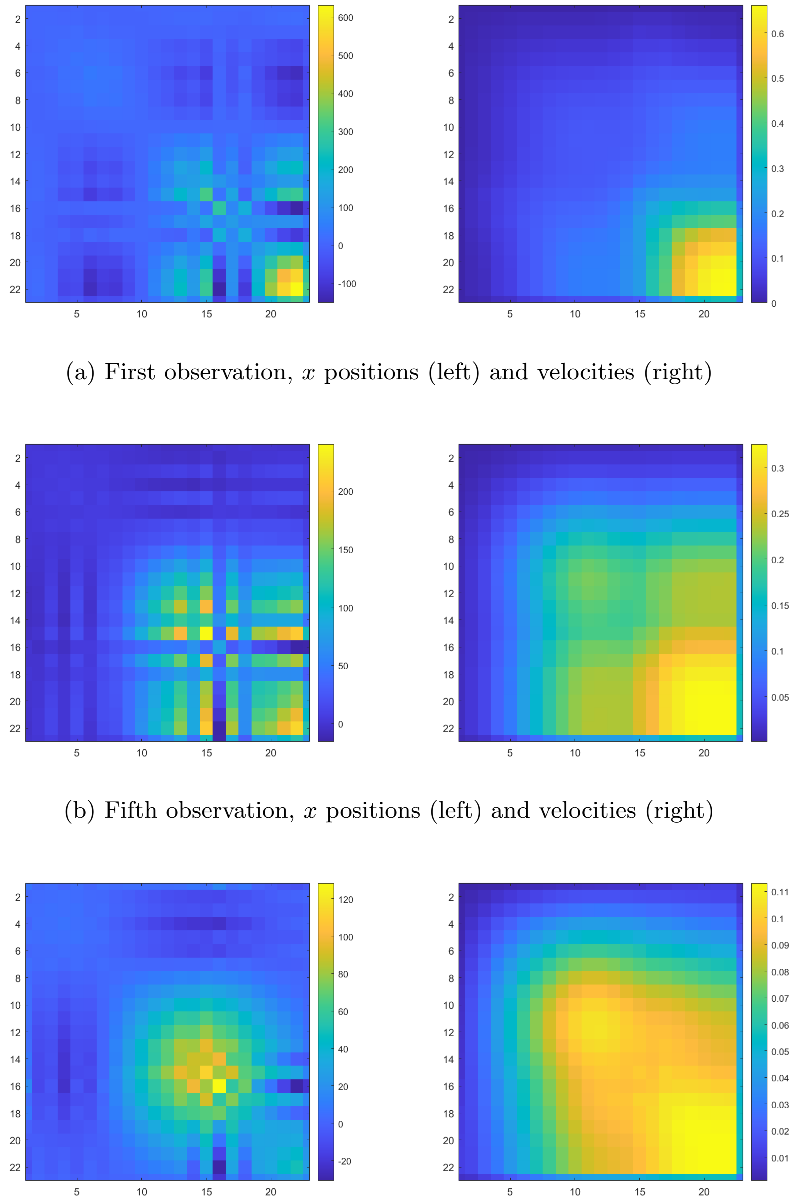
A rotation of this vector about the centerline yields the curvature normal vector, which can be projected onto the directors $\mathbf{d}_1^i$ and $\mathbf{d}_2^i$ to produce bending strains $(\kappa_1)_i$ and $(\kappa_2)_i$. To define a discrete twist, let $\mathbf{R}(\mathbf{t}^1, \mathbf{t}^2)$ be the unique rotation matrix that maps $\mathbf{t}_1$ to $\mathbf{t}^2$ and leaves $\mathbf{t}^1 \times \mathbf{t}^2$ unaltered. Then the discrete twist at vertex i is given by the angle τ_i between $\mathbf{d}_1^i$ and $\mathbf{R}(\mathbf{t}^{i-1}, \mathbf{t}^i)\mathbf{d}_1^{i-1}$, and can be determined form the relations

$$\begin{aligned} \cos(\tau_i) &= \mathbf{d}_1^i \cdot \mathbf{R}(\mathbf{t}^{i-1}, \mathbf{t}^i)\mathbf{d}_1^{i-1} \\ \sin(\tau_i) &= \mathbf{d}_2^i \cdot \mathbf{R}(\mathbf{t}^{i-1}, \mathbf{t}^i)\mathbf{d}_1^{i-1}. \end{aligned}$$

To integrate vertex based quantities over the rod, each vertex is assigned a resting length $\bar{l}_i = (\bar{l}^i + \bar{l}^{i-1})/2$. The three components of the discrete elastic energy is then expressed as

$$\begin{aligned} E_\kappa &= \frac{E}{2} \sum_i \|\boldsymbol{\kappa}_i - \bar{\boldsymbol{\kappa}}_i\|_{\mathbf{I}}^2 / \bar{l}_i, \\ E_\tau &= \frac{GJ}{2} \sum_i (\tau_i - \bar{\tau}_i)^2 / \bar{l}_i, \\ E_\epsilon &= \frac{EA}{2} \sum_i (l^i - \bar{l}^i)^2 / \bar{l}^i, \end{aligned}$$

and the total energy is the sum $E = E_\kappa + E_\tau + E_\epsilon$. From this elastic energy function, the forces and twisting moments acting on the rod can be derived analytically by taking partial derivatives with respect to the centerline positions $\mathbf{x}_i$ and the orientations of the directors $\mathbf{d}_1^i$. The Hessian of the elastic energy function can also be derived analytically, after some effort, and it becomes useful for quasi-static equilibrium problems and implicit time integration.



(a) First observation, x positions (left) and velocities (right)

(b) Fifth observation, x positions (left) and velocities (right)

(c) Tenth observation, x positions (left) and velocities (right)

Fig. 4: Components of the posterior precision that are directly attributable to each set of observations.