1. Inverse methods across various fields.

Across many fields, there exists usually something similar to the following.

$$parameters : m$$

$$data : d$$

$$map : f(m) \rightarrow d$$

This map is generated by the 'model' or the 'solution' F often given by a matrix or even a PDE like below.

$$\underbrace{F(u(x,t), m) = 0}_{u_t + m u_x = 0, \quad u(x,0) = u_0}$$

And what ends up happening usually is

$$\min_m ||f(m) - d||$$

A lot of incremental progress and research has been on improving optimization algorithms. However, I want to believe that no algorithm can capture the optimality that comes with framing a hard problem with the right perspective.

1.1. Bayes Rule

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

So, effectively

$$p(x|y) \propto p(y|x)p(x)$$

This is essentially the basis for Bayesian inverse problems. We want to find the parameters x that have generated the solution y. We have some initial information about x, this is our bias about x and that is encoded in the 'prior' $p(x)$. Also, the more likely the parameters are to generate the solution y, the more likely it is to be the true parameter. This is encoded by the likelihood function $p(y|x)$. So, maximising the likelihood is essentially the same as

$$\min_x ||y(x) - y_{true}||$$

So, a Bayesian inverse problem is the same as the standard minimzation with maybe a regularization term because that is our bias or our prior information about the problem. The difference is that Bayesian methods give you a distribution while minimization methods give you the minima.

For example, for Gaussian processes,

$$p(x) \propto e^{-\frac{(x-\mu_p)^2}{\sigma_p^2}}$$

$$p(y|x) \propto e^{-\frac{((x-y)^2}{\sigma_p 0^2}}$$

$$p(x|y) \propto e^{-\frac{((x-y)^2}{\sigma_p 0^2} - \frac{(x-\mu_p)^2}{\sigma_p^2}}$$

The standard deviation of $p(x|y)$ is lower than both σ_0, σ_p.

The posterior is easy to calculate in this 1-d case. But in general, the posterior might not have a closed form solution or it is very high-dimensional. In those cases, people use sampling like MCMC sampling to get an idea of the maxima. People also use variational methods to find the minimizer of the log-likeklihood.

2. The different approaches to estimation would all result in the same answer if everything was linear and Gaussian

2.1. BLUE

Suppose we want to measure something called X. You have some idea of what it should be like (your prior). And you have some observations. Based on these observations, you can get a good estimate of X. For example, think of wanting to estimate the temperature in your room and going around with a thermometer.

We have

$$Background(Prior) : x_b$$

$$Observation : y$$

$$Unknowntruth : x_t$$

$$Analysis(Posterior) : x_a$$

And we have errors defined by

$$e_b = x_b - x_t$$

$$e_o = y - x_t$$

$$e_a = x_a - x_t$$

We can make the following reasonable assumption that

$$e_b \perp e_o$$

We have the following rule/estimator

$$x_a = x_b + \theta(y - x_b)$$

$$x_a - x_t = x_b - x_t + \theta(y - x_t - (x_b - x_t))$$

$$e_a = e_b + \theta(e_o - e_b)$$

If $\mathbb{E}[e_o] = \mathbb{E}[e_o] = 0$

$$\mathbb{E}[e_a] = 0$$

So, the mean of the error of our analysis is zero. Let's look at the variance

$$\begin{aligned}
\mathbb{V}ar[e_a] &= \mathbb{E}[(e_a - \mathbb{E}[e_a])(e_a - \mathbb{E}[e_a])^T] \\
&= \mathbb{E}[e_a e_a^T] \\
&= \mathbb{E}[(e_b + \theta(e_o - e_b)(e_b + \theta(e_o - e_b))^T] \\
&= \mathbb{E}[e_b e_b^T + \theta^2(e_o e_o^T + e_b e_b^T) - 2\theta(e_b e_b^T)]
\end{aligned}$$

In the scalar case, this becomes

$$\sigma_a^2 = \sigma_b^2 + \theta^2(\sigma_o^2 + \sigma_b^2) - 2\theta\sigma_b^2$$

The θ that optimizes this is obtained by taking a derivative

$$\theta^* = \frac{\sigma_b^2}{\sigma_o^2 + \sigma_b^2}$$

So, our Best Linear Unbiased Estimator of x that minimizes the variance is given by

$$x_a = x_b + \frac{\sigma_b^2}{\sigma_o^2 + \sigma_b^2}(y - x_b)$$

If we had a lot of certainty in our bias, then $\sigma_b << \sigma_o$ and we have

$$x_a = x_b + 0 = x_b$$

If we were certain that our observation errors were negligible, then $\sigma_o << \sigma_b$ and we have

$$x_a = x_b + (y - x_b) = y$$

2.2. The optimization perspective

Let's try to frame this question differently. Suppose we had our background guess defined as

$$x_b = x_t + \epsilon_b, \quad \epsilon_b \sim N(0, B)$$

$$y = Hx_t + \epsilon_o, \quad \epsilon_o \sim N(0, R)$$

Here, B and R are the covariance matrices of the error in our bias and our observations. We can define the following cost function where the least squares is weighted with the covariance matrices.

$$J(x) = \frac{1}{2}(x - x_b)^T B^{-1}(x - x_b) + \frac{1}{2}(y - Hx)R^{-1}(y - Hx)$$

To find the minima, we need to take the derivative.

$$\nabla J(x)|_{x_{opt}} = 0$$

$$0 = B^{-1}(x_{opt} - x_b) - H^T R^{-1}(y - Hx_{opt})$$

$$\begin{aligned} x_{opt} &= x_b + (B^{-1} - H^T R^{-1} H)^{-1} H^T R^{-1}(y - Hx_b) \\ &= x_b + \underbrace{BH^T(R + HBH^T)^{-1}}_{K}(y - Hx_b) \end{aligned}$$

This K is our Kalman gain which tells you how much you have gained from your observations. If you look carefully at K, you can see how it is exactly θ^* in the scale case (and if H was the identity). So, a different perspective has unearthed the same idea of how our bias and our observations should be weighted.

2.3. Bayes rule

Suppose we wanted to estimate x which could be the temperature of a room. We expect x to be distributed normally around x_b.

$$x \sim N(x_b, \sigma_b^2)$$

We walk around the room making observations with our thermometer.

$$y = [y, y_2, \ldots, y_n]$$

$$y_i = x + \epsilon_o, \quad \epsilon_o = N(0, \sigma^2)$$

$$\begin{aligned} p(y|x) &= p(y_1|x)p(y_2|x)\ldots p(y_n|x) \\ &= \prod_{i=1}^{n} p(y_i|x) \\ &= \prod_{i=1}^{n} e^{-\frac{(y_i-x)^2}{2\sigma^2}} \\ &= e^{-\sum_i \frac{(y_i-x)^2}{2\sigma^2}} \end{aligned}$$

So, from Bayes rule,

$$\begin{aligned} p(x|y) &= p(y|x)p(x) \\ &= e^{-\sum_i \frac{(y_i-x)^2}{2\sigma^2} - \sum_i \frac{(x-x_b)^2}{2\sigma_b^2}} \\ &= N(\mu_{xy}, \sigma_{xy}) \end{aligned}$$

The product of two Gaussians is another Gaussian.

$$\begin{aligned} p(x|y) &= p(y|x)p(x) \\ &= e^{-\sum_i \frac{(y_i-x)^2}{2\sigma^2} - \sum_i \frac{(x-x_b)^2}{2\sigma_b^2}} \\ &= N(\mu_{xy}, \sigma_{xy}) \end{aligned}$$

and the mean of this Gaussian is given by

$$\mu_{x|y} = \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_b^2}\right)^{-1} \left(\sum \frac{y_i}{\sigma^2} + \frac{x_b}{\sigma_b^2}\right)$$

Taking $n = 1$ for simplicity, this can be written as

$$\begin{aligned} \mu_{x|y} &= \frac{\sigma^2\sigma_b^2}{\sigma^2 + \sigma_b^2}\left(\frac{y}{\sigma^2} + \frac{x_b}{\sigma_b^2}\right) \\ &= \frac{\sigma_b^2}{\sigma^2 + \sigma_b^2} y + \frac{\sigma^2}{\sigma^2 + \sigma_b^2} x_b \\ &= x_b + \frac{\sigma_b^2}{\sigma^2 + \sigma_b^2}(y - x_b) \end{aligned}$$

So, all roads lead to BLUE when things are linear and Gaussian.

3. Kalman Filter

3.1. Theory

The Kalman filter is a way to estimate the states of a process using sequential assimilation. The process

$$x_{k+1} = M_{k+1}x_k + w_k$$

with observations

$$y_k = H_k x_k + v_k$$

The noise is distributed by

$$w_k \sim \mathcal{N}(0, Q_k)$$
$$v_k \sim \mathcal{N}(0, R_k)$$

We define two of our estimates, one obtained by predicting x_k from x_{k-1} and another by correcting the prediction.

$$Forecast(Prediction) : x_k^f$$
$$Analysis(Correction) : x_k^a$$

It has errors defined by

$$e_k^f = x_k^f - x_k^t$$
$$e_k^a = x_k^a - x_k^t$$

where x^t refers to the truth.

And covariance matrices given by

$$P_k^f = cov(e_k^f) = \mathbf{E}[e_k^f (e_k^f)^T]$$
$$P_k^a = cov(e_k^a) = \mathbf{E}[e_k^a (e_k^a)^T]$$

We have our Kalman Filter correction rule.

$$x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$$

where K_k is the Kalman gain at step k.

$$\begin{aligned} x_k^a &= x_k^f + K_k(Hx_k^t + v_k - H_k x_k^f) \\ &= x_k^f + K_k(H(x_k^t - x_k^f) + v_k) \end{aligned}$$

$$\begin{aligned} e_k^a = x_k^a - x_t &= x_k^f + K_k(Hx_k^t + v_k - H_k x_k^f) \\ &= x_k^f + K_k(H(x_k^t - x_k^f) + v_k) - x_k^t \\ &= K_k(v_k - He_k^f) - e_k^f \end{aligned}$$

$$P_k^a = \mathbf{E}[e_k^a(e_k^a)^T] = \mathbf{E}[(K_k(v_k - He_k^f) - e_k^f)(K_k(v_k - He_k^f) - e_k^f)^T]$$
$$= (I - K_kH_k)P_k^f(I - K_kH_k)^T + K_kR_kK_k^T$$

$$P_k^a = (I - K_kH_k)P_k^f(I - K_kH_k)^T + K_kR_kK_k^T$$

To make the effect of the covariance matrix as minimal as possible, we try to minimize its trace using these helpful matrix derivatives.

$$\frac{d}{dA}Tr(AB) = B^T$$

$$\frac{d}{dA}Tr(ACA^T) = 2AC$$

$$\frac{d}{dK_k}Tr(P_k^a) = -2(H_kP_k^f)^T + 2K_k(H_kK_kH_k^T + R_k)$$

This helps us reveal the Kalman gain that minimizes the trace of the covariance.

$$K_k = P_k^fH_k^T(H_kP_k^fH_k^T + R_k)^{-1}$$

And thus we get the minimized covariance of our analysis.

$$P_k^a = (I - K_kH_k)P_k^f$$

So, we have the Kalman procedure

Prediction

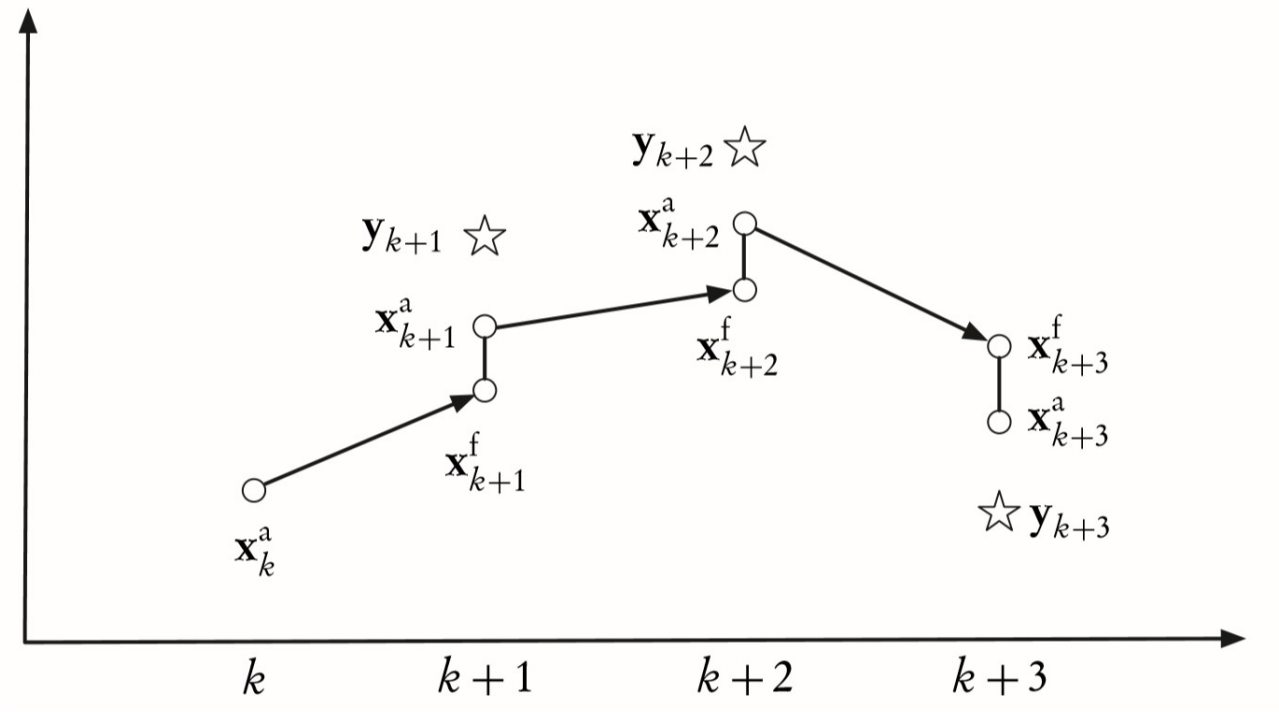
$$x_{k+1}^f = M_{k+1}x_k^a$$
$$P_{k+1}^f = M_{k+1}P_k^aM_{k+1}^T + Q_{k+1}$$

Correction

$$K_{k+1} = P_{k+1}^fH^T(HP_{k+1}^fH^T + R_{k+1})^{-1}$$

$$x_{k+1}^a = x_{k+1}^f + K_{k+1}(y_{k+1} - Hx_{k+1}^f)$$

$$P_{k+1}^a = (I - K_{k+1}H)P_{k+1}^f$$



3.2. A linear example

This is an example which use the Kalman filter to find the velocity of a car when only the position is observed.

The process is given by

$$u = u_0 + vt$$

And the state is defined by

$$x = \begin{bmatrix} u \\ v \end{bmatrix}$$

We have the following map

$$x_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k$$

And the following observation matrix.

$$H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Using this, the Kalman filter converges quickly to the true velocity.

One thing to remember is that the Kalman filter is optimal for linear cases but non-linear cases can fail. So, people use Extended KF and Ensemble KF which is very state of the art.

3.3. A nonlinear example

Suppose we want to estimate the dynamics of a pendulum where its position u is given by

$$u(t) = \sin(\theta t) + \epsilon$$

So, we write as a differential equation

$$u' = \theta \cos(\theta t) + \epsilon_t$$

After linearizing around θ_0

$$u' = \theta_0 \cos(\theta_0 t) + (\cos(\theta_0 t) - \theta_0 \sin(\theta_0 t))(\theta - \theta_0) + \epsilon_t$$

where ϵ is the model error and θ is an unknown parameter. So, we use the Kalman filter with

$$x = \begin{bmatrix} u \\ \theta \end{bmatrix}$$

Following the Kalman filter notation, we have

$$x^f_{k+1} = M_{k+1}x^a_k + c$$

where c is an offset vector needed in the case where we do a taylor expansion around $\theta_0 \neq 0$.

$$M_{k+1} = \begin{bmatrix} 1 & \Delta t(-\theta_0 \sin(\theta_0 t) + \cos(\theta_0 t)) \\ 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} \Delta t \theta_0 t_k sin(\theta t_k) \\ 0 \end{bmatrix}$$

If $\theta_0 = 0$,

$$M_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Then, we have

$$P^f_{k+1} = M_{k+1}P^a_k M^T_{k+1} + Q$$

where we choose

$$P^a_0 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_u & 0 \\ 0 & 0.0001 \end{bmatrix}$$

After that, we seek to incorporate the measurements y which have the following relation with x

$$y = Hx + R$$

For our case, since we cannot measure θ, we have

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, R = \begin{bmatrix} R_u & 0 \\ 0 & 0.001 \end{bmatrix}$$

Then, use the update rules that minimize the variance of the unbiased linearized analysis.

$$K_{k+1} = P^f_{k+1}H^T(HP^f_{k+1}H^T + R)^{-1}$$

$$x^a_{k+1} = x^f_{k+1} + K_{k+1}(y - Hx^f_{k+1})$$

$$P^a_{k+1} = (I - K_{k+1}H)P^f_{k+1}$$

4. Variational approach to inverse problems

The calculus of variations is an old reliable field that has applications in inverse problems especially when differential equations are used.

Suppose we have a system obeying this law (pde/ode etc).

$$L(u, m) = f$$

Now, we want to find the parameters m from observations. So, we define a cost function

$$J(m) = \int \underbrace{(u(m) - u_{obs})^2}_{J_1} d\Omega$$

Now, we start with a guess for m, compute the gradient and go down the gradient hill till you reach the minima. But we need the gradient for that.

$$\begin{aligned}\delta J &= \nabla_m J \delta m \\ &= \nabla_u J \frac{du}{dm} \delta m \\ &= \nabla_u J \delta u \\ &= < \nabla_u J_1 \delta u >\end{aligned}$$

δu could be very expensive to compute numerically because you have to see the change in u spread possibly in space and time for a small change in m. Calculus of variation helps up reformulate the gradient into a simpler form.

Take our law.

$$L(u, m) = f$$

Its variation should be zero.

$$\delta L = \nabla_u L \delta u + \nabla_m L \delta m = 0$$

Now, I multiply the above expression with a test function p and integrate.

$$< p \nabla_u L \delta u > + < p \nabla_m L \delta m >= 0$$

Adding this to δJ,

$$\begin{aligned}\nabla_m J \delta m &= < \nabla_u J_1 \delta u > + < p \nabla_u L \delta u > + < p \nabla_m L \delta m > \\ &= < \delta u (\nabla_u J_1 + \nabla_u L^* p) > + < \delta m \nabla_m L^* p >\end{aligned}$$

In the last step, we used the definition of adjoint

$$< p \nabla_u L \delta u >=< \delta u \nabla_u L^* p >, \quad < \delta m \nabla_m L^* p >=< p \nabla_m L \delta m >$$

Now, if $\nabla_u L^* p = -\nabla_u J_1$, then we have

$$\nabla_m J \delta m =< \delta m \nabla_m L^* p >$$

This allows you to find the gradient $\nabla_m J$ a lot easier.
Great. Why don't we do this always? Couple of difficulties:

- It's not always possible to compute the adjoint especially with boundary or initial conditions that do not add up
- There might be multiple non-unique minima. People try to solve this through regularization and hybrid algorithms.
- The cost function can sometimes be very flat. This can be avoided by proper scaling of parameters [Nocedal and Wright]
- The gradient obtained is for the continuous problem while numerically we never do the continuous problem.

Important: In these notes, the technique used to find the continuous adjoint is the Lagrange multiplier approach. An alternate approach is the Tangent Linear Model method which is discussed in the first reference book.

4.1. An ODE example

Suppose we have the equation

$$-(a(x)u'(x))' - u'(x) = q(x), \quad 0 < x < 1$$
$$u(0) = 0, \quad u(1) = 0$$

We can define a cost function as

$$J[a] = \frac{1}{2}\int_0^1 (u - u_{obs})^2 dx + \int_0^1 p(-(au')' - u' - q)dx$$

$$\delta J[a] = \int_0^1 (u - u_{obs})\delta u dx + \int_0^1 \delta p(-(au')' - u' - q)dx + \int_0^1 p(-(\delta a u' + a\delta u')' - \delta u')dx$$

$$\begin{aligned}\delta J[a] &= \int_0^1 (u - u_{obs})\delta u dx + \int_0^1 p(-(\delta a u' + a\delta u')' - \delta u')dx \\ &= \int_0^1 (u - u_{obs})\delta u dx + \int_0^1 p'(\delta a u' dx + a\delta u' dx) + p'\delta u dx \\ &= \int_0^1 (u - u_{obs} + p' - (ap')')\delta u dx + \int_0^1 p'u'\delta a dx\end{aligned}$$

We used integration by parts above and omitted the boundary terms for clarity. The boundary terms are zero if p is zero on the boundary. Also, I think $p'a\delta u$ should be zero on the boundary. (Later I realized that that term implied that $a(0) = a(1) = 0$).

If p satisfies

$$p' - (ap')' = -(u - u_{obs})$$
$$p(0) = 0, \quad p(1) = 0$$

then

$$\delta J[a] = \int_0^1 p'u'\delta a dx$$

And we obtain our gradient

$$\nabla_{a(x)} J[a] = p'u'$$

4.2. Constant coefficient case

In the previous example, what happens when $a(x)$ is a constant a?

$$J[a] = \frac{1}{2}\int_0^1 (u - u_{obs})^2 dx + \int_0^1 p(-au'' + u' - q)dx$$

$$\begin{aligned}
\delta J[a] &= \int_0^1 (u - u_{obs})\delta u dx + \int_0^1 p(-\delta a u'' - a\delta u'' - \delta u')dx \\
&= \int_0^1 (u - u_{obs})\delta u dx + \int_0^1 -pu''\delta a dx - p''a\delta u dx + p'\delta u dx \\
&= \int_0^1 (u - u_{obs} + p' - ap'')\delta u dx - \int_0^1 pu''\delta a dx
\end{aligned}$$

Our adjoint equation is

$$p' - (ap')' = -(u - u_{obs})$$
$$p(0) = 0, \quad p(1) = 0$$

and we have

$$\delta J[a] = -\int_0^1 pu''\delta a dx$$

And we obtain our gradient

$$\nabla_a J[a] = -\int_0^1 pu'' dx$$

You might face some confusion because this is of a different form from the gradient in the previous section especially because of the presence of an integral. What may ease your confusion is thinking of the integral as a summation you are differentiating. In this case, a is constant over the summation so we can take it out of the sum leaving us still with a sum i.e an integral.

4.3. Linear PDE example

Suppose we have the heat equation

$$\frac{\partial u}{\partial t} - \nabla.(\nu \nabla u) = 0$$
$$u(x, 0) = u_0(x), \quad u(0, t) = 0, u(L, t) = \eta(t)$$

Each one of these unknown parameters corresponds to classical inverse problems.

- internal control $\nu(x)$ - tomography.
- initial control $u_0(x)$ - source detection problem.
- boundary control $\eta(t)$

We can solve for all 3 simultaneously with this cost function.

$$J[\nu, u_0, \eta] = \frac{1}{LT}\int_0^T \int_0^L (u - u_{obs})^2 dxdt + \frac{1}{LT}\int_0^T \int_0^L p(u_t - (\nu u_x)_x)dxdt$$

After integration by parts, we get the following adjoint equation

$$\frac{\partial p}{\partial t} - \nabla.(\nu \nabla p) = 2(u - u_{obs})$$
$$p(0,t) = 0, \quad p(L,t) = 0, \quad p(x,T) = 0$$

The adjoint equation is solved backwards in time and we can compute the following gradients.

$$\nabla_{\nu(x)} = \frac{1}{T}\int_0^T u_x p_x dt$$
$$\nabla_{u|_{t=0}} = -p|_{t=0}$$
$$\nabla_{\eta|_{x=L}} = \nu p|_{x=L}$$

5. 3d-Var and 4d-Var

We have two ways of doing things

- **AtD**: Adjoint then Discretize
- **DtA**: Discretize then Adjoint

5.1. The stationary case: 3d-Var

$$x^b = x^t + \epsilon^b$$
$$y = Hx^t + \epsilon^o$$

We start with a cost function of the following form:

$$J(x) = \frac{1}{2}(x - x^b)^T B^{-1}(x - x^b) + \frac{1}{2}(Hx - y)^T R^{-1}(Hx - y)$$

$$x^a = x^b + K(y - H(x^b))$$

where

$$K = BH^T(HBH^T + R)^{-1}$$

Now, in real life applications like a PDE or ODE, the matrices used to compute K could be way too large to store in memory.

5.2. The non-stationary case: 4d-Var

$$y_k = H_k x_k + \epsilon_k^o$$

And let us assume our model is devoid of any error.

$$x_{k+1} = M_{k+1} x_k$$

Then a single variable, the initial condition x_0 determines all of the x_k. We have the following cost function.

$$J(x_0) = \frac{1}{2}(x_0 - x_0^b)^T (P_0^b)(x_0 - x_0^b) + \frac{1}{2}\sum_{k=0}^{K}(H_k x_k - y_k)^T R_k^{-1}(H_k x_k - y_k)$$

In the presence of model uncertainty,

$$x_{k+1}^t = M_{k+1}x_k^t + \eta_{k+1}$$

$$\begin{aligned} J(x_0, x_1, \ldots, x_K) = \frac{1}{2}(x_0 - x_0^b)^T (P_0^b)(x_0 - x_0^b) &+ \frac{1}{2}\sum_{k=0}^{K}(H_k x_k - y_k)^T R_k^{-1}(H_k x_k - y_k) \\ &+ \frac{1}{2}\sum_{k=0}^{K-1}(x_{k+1} - M_{k+1}x_k)^T Q_{k+1}^{-1}(x_{k+1} - M_{k+1}x_k) \end{aligned}$$

5.3. Adjoint approach to 4d-Var

We have the cost

$$J(x_0) = \frac{1}{2}(x_0 - x_0^b)^T (P_0^b)^{-1}(x_0 - x_0^b) + \frac{1}{2}\sum_{k=0}^{K}(H_k x_k - y_k)^T R_k^{-1}(H_k x_k - y_k)$$

$$\delta J = (\nabla_{x_0} J)^T \delta x_0$$

and the dynamics

$$\begin{aligned} x_{k+1} &= M_{k+1}x_k \\ \delta x_{k+1} &= M_{k+1}\delta x_k \\ \delta x_{k+1} - M_{k+1}\delta x_k &= 0 \\ p_{k+1}^T(\delta x_{k+1} - M_{k+1}\delta x_k) &= 0 \end{aligned}$$

$$\delta J = (x_0 - x_0^b)^T (P_0^b)^{-1}\delta x_0 + \sum_{k=0}^{K}(H_k x_k - y_k)^T R_k^{-1} H_k \delta x_k$$

Adding the above two lines we get

$$\delta J = (x_0 - x_0^b)^T (P_0^b)^{-1}\delta x_0 + \sum_{k=0}^{K}(H_k x_k - y_k)^T R_k^{-1} H_k \delta x_k - \sum_{k=0}^{K-1} p_{k+1}^T(\delta x_{k+1} - M_{k+1}\delta x_k)$$

$$\begin{aligned} \delta J = [(P_0^b)^{-1}(x_0 - x_0^b) + H_0^T R^{-1}(H_0 x_0 - y_0) + M_1^T p_1]^T \delta x_0 \\ + \sum_{k=1}^{K-1}[H_k^T R^{-1}(H_k x_k - y_k) - p_k - M_{k+1}^T p_{k+1}]^T \delta x_k \\ + [H_K^T R^{-1}(H_K x_K - y_K) - p_K]^T \delta x_K \end{aligned}$$

To kill some terms, we do the following in successive order.

$$p_K = H_K^T R^{-1}(H_K x_K - y_K)$$
$$p_k = H_k^T R^{-1}(H_k x_k - y_k) - M_{k+1}^T p_{k+1}, \quad k = K-1, \dots 1$$
$$p_0 = (P_0^b)^{-1}(x_0 - x_0^b) + H_0^T R^{-1}(H_0 x_0 - y_0) + M_1^T p_1$$

So, we have

$$\delta J = (\nabla_{x_0} J)^T \delta x_0 = p_0^T \delta x_0$$

And hence,

$$\nabla_{x_0} J = p_0$$

5.4. Extensions

There are practical variants of 3d-Var and 4d-Var. Some are

- Incremental 3D-Var and 4D-Var
- FGAT 3d Var (First guess at appropriate time)

There are also various extensions.

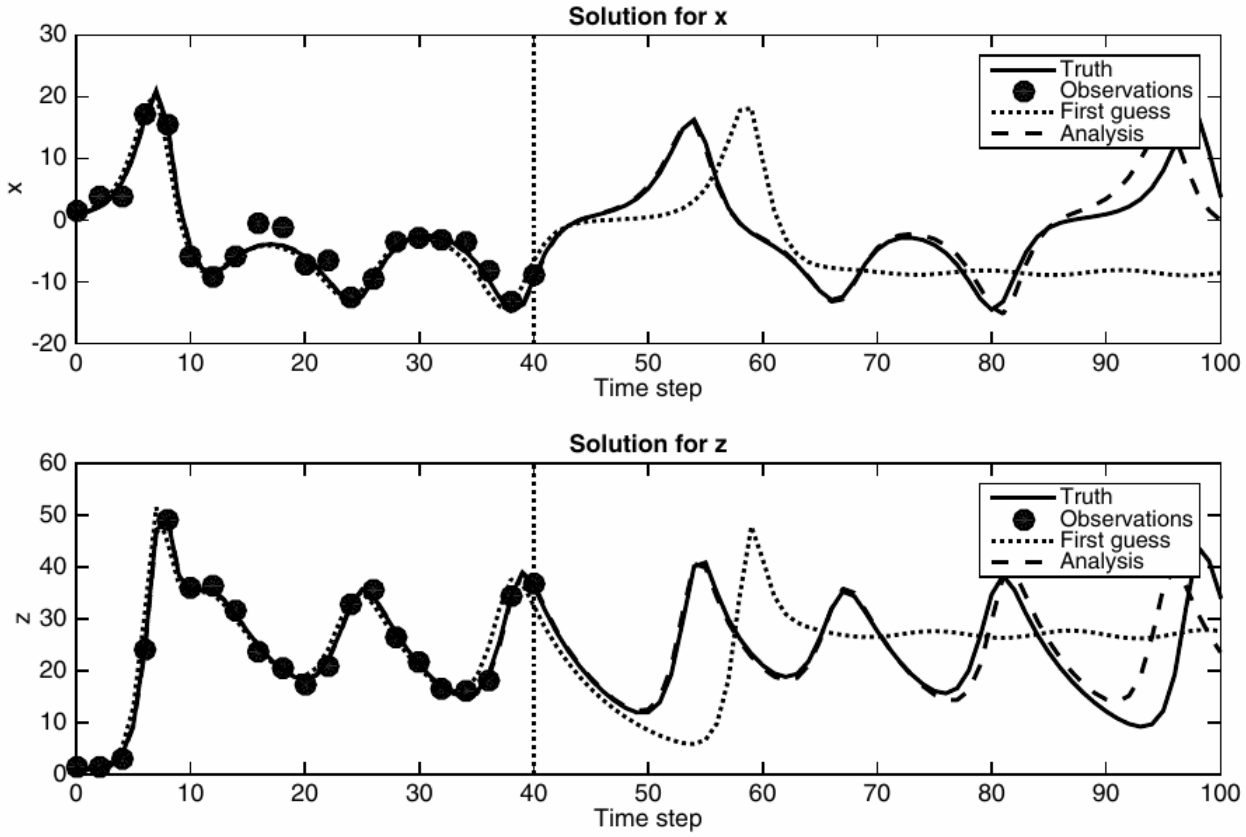
- Parameter estimation.
- Preconditioning
- Covariance matrix modeling
- Model error

$$\frac{dx}{dt} = M(x) + \eta(t)$$
$$J(x_0, \eta) = \frac{1}{2}||x_0 - x^b||^2 \frac{1}{2}\int_0^T ||y(t) - H(x)||^2 dt + \frac{1}{2}\int_0^T ||\eta(t)||^2 dt$$

5.5. 4d-Var applied to the Lorenz system

$$\frac{dx}{dt} = -\sigma(x-y),$$
$$\frac{dy}{dt} = \rho x - y - xz,$$
$$\frac{dz}{dt} = xy - \beta z,$$

- Lorenz system is chaotic.
- We want to estimate the initial condition from observations.
- True initial condition is (1,1,1)
- Initial guess is (1.2,1.2,1.2)



6. References

- Data Assimilation: Methods, Algorithms, and Applications by Marc Bocquet, Mark Asch, and Maëlle Nodet:

  This book is a really strong consolidated collections of inverse methods. But it has a lot of mistakes especially in the derivations. The variational examples in these can be found in Chapter 2 of this book.

- Dynamic Data Assimilation: A Least Squares Approach by John M. Lewis, S. Lakshmivarahan, and Sudarshan Kumar Dhall

  This is probably a more standard book but it doesn't have a wide breadth of topics and gets into the details too much for my taste.

- Physics-based covariance models for Gaussian processes with multiple outputs Emil M. Constantinescu, Mihai Anitescu

  People interested in deriving equations for how the covariance of your physical system propagates should check this out.

- PETSc TSAdjoint: a discrete adjoint ODE solver for first-order and second-order sensitivity analysis

  Section 2 of this paper shows the process of deriving the discrete adjoint clearly.