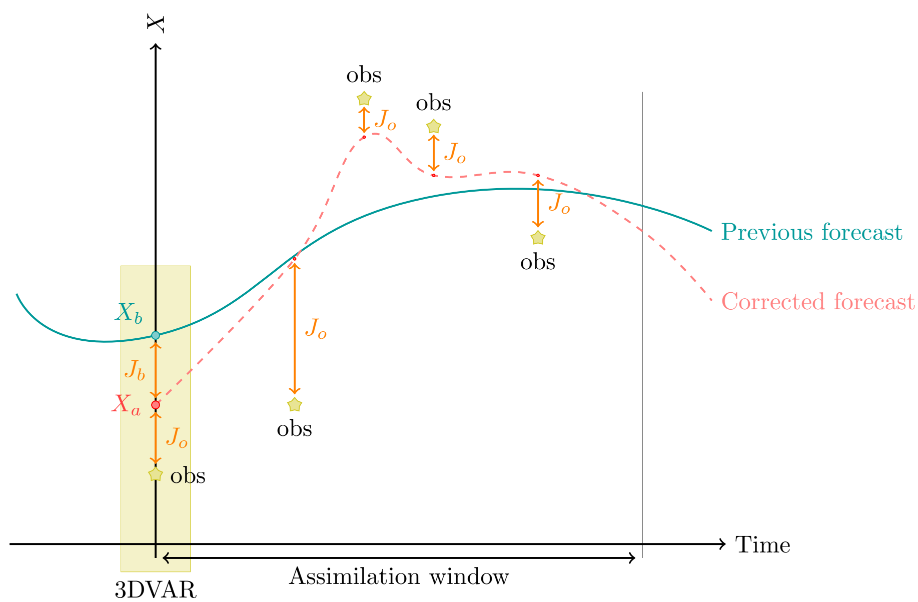
DA - Adjoint Method

Mark Asch - CSU/IMU/2023



Outline of the course (I)

Adjoint methods and variational data assimilation (4h)

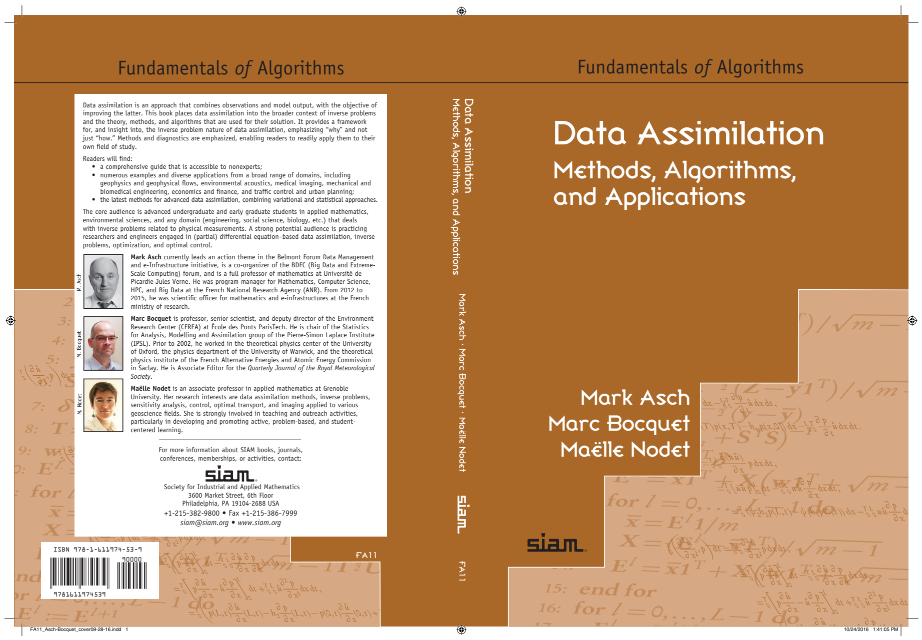
1. Introduction to data assimilation: setting, history, overview, definitions.
2. Adjoint method.
3. Variational data assimilation methods:
   (a) 3D-Var,
   (b) 4D-Var.

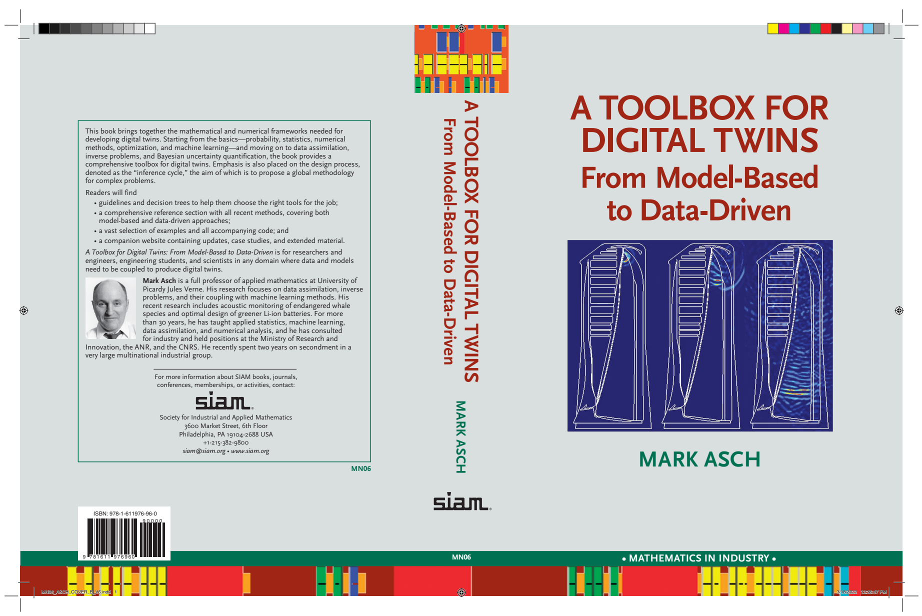
Outline of the course (II)

Statistical estimation, Kalman filters and sequential data assimilation (4h)

1. Introduction to statistical DA.
2. Statistical estimation.
3. The Kalman filter.
4. Nonlinear extensions and ensemble filters.

Reference Textbooks



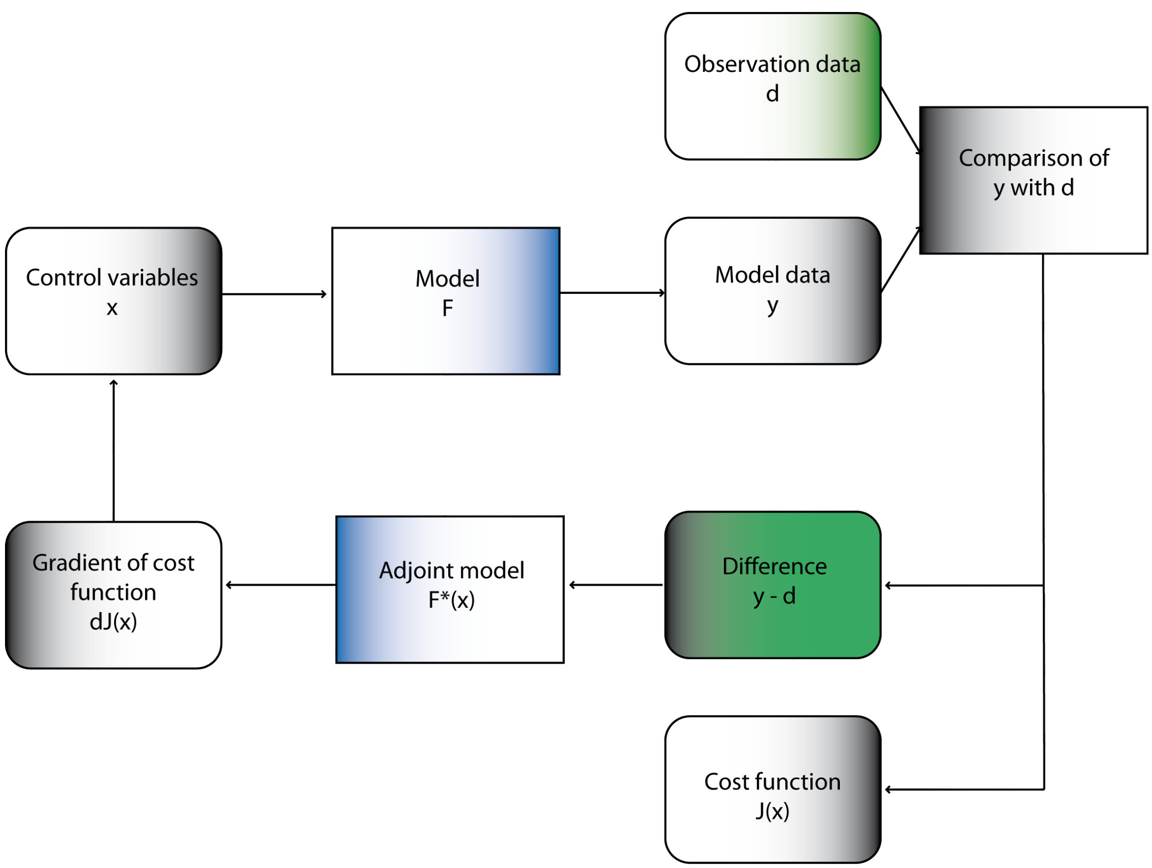


ADJOINT METHOD FOR INVERSE PROBLEMS

Adjoint Methods (I)

- A very general approach for solving inverse problems... including Machine Learning!

- Variational DA is based on an adjoint approach.



Adjoint Methods (II) - definition

Definition 1. *An* **adjoint method** *is a general mathematical technique, based on variational calculus, that enables the computation of the gradient of an objective, or cost functional with respect to the model parameters in a very efficient manner.*

Adjoint Methods (III) - continuous formulation

- Let $\mathbf{u}(\mathbf{x},t)$ be the state of a *dynamical system* whose behavior depends on model parameters $\mathbf{m}(\mathbf{x},t)$ and is described by a differential operator equation

$$\mathbf{L}(\mathbf{u},\mathbf{m}) = \mathbf{f},$$

  where $\mathbf{f}(\mathbf{x},t)$ represents external forces.

- Define a *cost function* $J(\mathbf{m})$ as an energy functional[1] or, more commonly, as a misfit functional that quantifies the error (L^2-distance[2]) between the observation and the model prediction $\mathbf{u}(\mathbf{x},t;\mathbf{m})$.

[1] A functional is a generalization of a function. The functional depends on functions, whereas a function depends on variables. We then say that a functional is mapping from a space of functions into the real numbers.

[2] The L^2-space is a Hilbert space of (measurable) functions that are square-integrable (in Lebesgue sense).

For example,

$$J(\mathbf{m}) = \int_0^T \int_\Omega \left(\mathbf{u}(\mathbf{x}, t; \mathbf{m}) - \mathbf{u}^{\mathrm{obs}}(\mathbf{x}, t)\right)^2 \, \mathrm{d}\mathbf{x}\mathrm{d}t,$$

where $\mathbf{x} \in \Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$, and $0 \le t \le T$.

- Our objective is to choose the model parameters $\mathbf{m}$ as a function of the observed output $\mathbf{u}^{\mathrm{obs}}$, such that the cost function $J(\mathbf{m})$ is minimized.

- The minimization is most frequently performed by a gradient-based method, the simplest of which is steepest gradient, though usually some variant of a quasi-Newton approach is used—see [Asch2022, Nocedal2016].

- If we can obtain an expression for the gradient, then the minimization will be considerably facilitated.

- This is the objective of the adjoint method that provides an **explicit formula for the gradient of $J(\mathbf{m})$.**

Adjoint Methods (IV) - optimization formulation

- Suppose we are given a (P)DE,

$$F(\mathbf{u}; \mathbf{m}) = 0, \tag{1}$$

where

⇒ $\mathbf{u}$ is the state vector,
⇒ $\mathbf{m}$ is the parameter vector, and
⇒ F includes the partial differential operator $\mathbf{L}$, the right-hand side (source) $\mathbf{f}$, boundary and initial conditions.

- Note that the components of $\mathbf{m}$ can appear as any combination of

⇒ coefficients in the equation,
⇒ the source,

⇒ or as components of the boundary/initial conditions.

- To solve this very general parameter estimation problem, we are given a cost function $J(\mathbf{m};\mathbf{u})$.

  ⇒ The constrained optimization problem is then

$$\begin{cases} \text{minimize}_{\mathbf{m}} & J\left(\mathbf{u}(\mathbf{m}),\mathbf{m}\right) \\ \text{subject to} & F(\mathbf{u};\mathbf{m}) = 0, \end{cases}$$

  where J can depend on both $\mathbf{u}$ and on $\mathbf{m}$ explicitly in the presence of eventual regularization terms.

  ⇒ Note:

  → the constraint is a (partial) differential equation and

  → the minimization is with respect to a (vector) function.

  ⇒ This type of optimization is the subject of variational calculus, a generalization of classical calculus where differentiation is performed with respect to a variable, not a function.

- The gradient of J with respect to $\mathbf{m}$ (also known as the *sensitivity*) is then obtained by the chain-rule,

$$\nabla_{\mathbf{m}} J = \frac{\partial J}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{m}} + \frac{\partial J}{\partial \mathbf{m}}.$$

  $\Rightarrow$ The partial derivatives of J with respect to $\mathbf{u}$ and $\mathbf{m}$ are readily computed from the expression for J,

  $\Rightarrow$ but the derivative of $\mathbf{u}$ with respect to $\mathbf{m}$ requires a potentially very large number of evaluations, corresponding to the product of the dimensions of $\mathbf{u}$ and $\mathbf{m}$ that can both be very large.

- The adjoint method is a way to avoid calculating all of these derivatives.

- We use the fact that if $F(\mathbf{u}; \mathbf{m}) = 0$ everywhere, then this implies that the total derivative of F with respect to $\mathbf{m}$ is equal to zero everywhere too.

- Differentiating the PDE (1), we can thus write

$$\frac{\partial F}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{m}} + \nabla_{\mathbf{m}} F = 0.$$

- This can be solved for the untractable derivative of $\mathbf{u}$ with respect to $\mathbf{m}$, to give

$$\frac{\partial \mathbf{u}}{\partial \mathbf{m}} = -\left(\frac{\partial F}{\partial \mathbf{u}}\right)^{-1} \nabla_{\mathbf{m}} F$$

assuming that the inverse of $\partial F / \partial \mathbf{u}$ exists.

- Substituting in the expression for the gradient of J, we obtain

$$\begin{aligned} \nabla_{\mathbf{m}} J &= -\frac{\partial J}{\partial \mathbf{u}}\left(\frac{\partial F}{\partial \mathbf{u}}\right)^{-1} \nabla_{\mathbf{m}} F + \frac{\partial J}{\partial \mathbf{m}}, \\ &= \mathbf{p} \nabla_{\mathbf{m}} F + \frac{\partial J}{\partial \mathbf{m}} \end{aligned}$$

where

$\Rightarrow$ we have conveniently defined $\mathbf{p}$ as the solution of the *adjoint equation*

$$\left(\frac{\partial F}{\partial \mathbf{u}}\right)^{\mathrm{T}} \mathbf{p} = -\frac{\partial J}{\partial \mathbf{u}}.$$

Adjoint Methods (V) - summing up

In summary, we have a three-step procedure that combines a model-based approach (through the PDE) with a data-driven approach (through the cost function):

1. Solve the *adjoint equation* for the adjoint state, $\mathbf{p}$.

2. Using the adjoint state, compute the *gradient* of the cost function J.

3. Using the gradient, solve the *optimization* problem to estimate the parameters $\mathbf{m}$ that *minimize the mismatch* between model and observations.

This key result enables us to compute the desired gradient, $\nabla_{\mathbf{m}} J$, without the explicit knowledge of the variation of $\mathbf{u}$.

- A number of important remarks can be made.

  1. We obtain *explicit* formulas—in terms of the adjoint state—for the gradient with respect to each/any model parameter. Note that this has been done in a completely general setting, without any restrictions on the operator F or on the model parameters $\mathbf{m}$.
  2. The *computational cost* is one solution of the adjoint equation which is usually of the same order as (if not identical to) the direct equation, but with a reversal of time. Note that for nonlinear equations this may not be the case and one may require four or five times the computational effort.
  3. The *variation* (Gâteaux derivative) of F with respect to the model parameters $\mathbf{m}$ is, in general, straightforward to compute.
  4. We have not explicitly considered boundary (or initial) conditions in the above, general approach. In real cases, these are potential sources of difficulties for the use of the adjoint approach—the *discrete adjoint* can provide a way

to overcome this hurdle.

5. For complete mathematical rigor, the above development should be performed in an appropriate *Hilbert space* setting that guarantees the existence of all the inner products and adjoint operators. The interested reader could consul [Troltzsch2010].
6. In many real problems, the optimization of the misfit functional leads to *multiple local minima* and often to very "flat" cost functions. These are hard problems for gradient-based optimization methods. These difficulties can be (partially) overcome by a panoply of tools:
   (a) *Regularization* terms can alleviate the non-uniqueness problem.
   (b) *Rescaling* the parameters and/or variables in the equations can help with the "flatness." This technique is often employed in numerical optimization.
   (c) *Hybrid* algorithms, that combine stochastic and deterministic optimization (e.g., Simulated Annealing), can be used to avoid local

minima.

(d) Judicious use of machine learning techniques and methods.

7. When measurement and modeling errors can be modeled by Gaussian distributions and a background (prior) solution exists, the objective function may be generalized by including suitable *covariance matrices*. This is the approach employed systematically in data assimilation—see below.

Adjoint Methods (VI) - Examples

- We will now present a series of examples where we apply the adjoint approach to increasingly complex cases of inverse problems, for which we have

  $\Rightarrow$ a cost/mismatch function to minimze
  $\Rightarrow$ subject to the constraint of a (P)DE.

- There are two alternative methods for the derivation of the adjoint equation:

  $\Rightarrow$ the Lagrange multiplier approach
  $\Rightarrow$ and the tangent linear model (TLM) approach.

- After seeing the two in action, the reader can adopt the one that suits her/him best.

- Note that the Lagrangian approach supposes that we perturb the sought for parameters and is thus not applicable to inverting for constant-valued parameters, in which case we must resort to the tangent linear model approach.

ADJOINT METHOD EXAMPLES

Example: TLM Adjoint Method for Parameter Identification

- Consider the parameter identification problem for the 1D convection-diffusion equation,

$$\begin{cases} -bu''(x) + cu'(x) = f(x), \quad 0 < x < 1, \\ u(0) = 0, \ u(1) = 0, \end{cases} \tag{2}$$

where

⇒ f is a given function
⇒ b and c are the unknown (constant) parameters that we seek to identify
⇒ using observations of $u(x)$ on $[0, 1]$.

- The least-squares error cost function is

$$J(b, c) = \frac{1}{2} \int_0^1 \left(u(x) - u^{\mathrm{obs}}(x) \right)^2 \mathrm{d}x.$$

- We calculate its gradient by introducing the *tangent linear model*.

⇒ Perturbing the cost function by a small perturbation in the direction α with respect to the two parameters gives

$$J(b+\alpha\delta b, c+\alpha\delta c) - J(b,c)$$
$$= \frac{1}{2}\int_0^1 \left(\tilde{u} - u^{\mathrm{obs}}\right)^2 - \left(u - u^{\mathrm{obs}}\right)^2 \mathrm{d}x,$$

where

→ $\tilde{u} = u_{b+\alpha\delta b, c+\alpha\delta c}$ is the perturbed solution and

→ $u = u_{b,c}$ is the unperturbed one.

⇒ Expanding and rearranging, we obtain

$$J(b+\alpha\delta b, c+\alpha\delta c) - J(b,c)$$
$$= \frac{1}{2}\int_0^1 \left(\tilde{u} + u - 2u^{\mathrm{obs}}\right)\left(\tilde{u} - u\right) \mathrm{d}x.$$

⇒ Now we divide by α on both sides of the equation and pass to the limit $\alpha \to 0$, to obtain the directional derivative (which is the derivative with respect to the parameters, in the direction of the perturbations),

$$\hat{J}\left[b,c\right]\left(\delta b,\delta c\right) = \int_0^1 \left(u - u^{\mathrm{obs}}\right)\hat{u}\,\mathrm{d}x, \quad (3)$$

where

→ we have defined

$$\hat{u} = \lim_{\alpha\to 0}\frac{\tilde{u}-u}{\alpha},$$
$$\hat{J}\left[b,c\right]\left(\delta b,\delta c\right) = \lim_{\alpha\to 0}\frac{J(b+\alpha\delta b, c+\alpha\delta c) - J(b,c)}{\alpha}$$

→ and we have moved the limit under the integral sign.

⇒ Now use this definition to find the equation satisfied by $\hat{u}$.

→ We have the perturbed equation,

$$\begin{cases} -(b+\alpha\delta b)\tilde{u}'' + (c+\alpha\delta c)\tilde{u}' = f, \\ \tilde{u}(0) = 0,\ \tilde{u}(1) = 0, \end{cases}$$

and the given model (2),

$$\begin{cases} -bu'' + cu' = f, \\ u(0) = 0,\ u(1) = 0. \end{cases}$$

$\rightarrow$ Then, subtracting these two equations and passing to the limit (using the definition of $\hat{u}$) we obtain

$$\begin{cases} -b\hat{u}'' - (\delta b)u'' + c\hat{u}' + (\delta c)u' = 0, \\ \hat{u}(0) = 0,\ \hat{u}(1) = 0. \end{cases}$$

$\Rightarrow$ Regrouping terms, we can now define the *tangent linear model*

$$\begin{cases} -b\hat{u}'' + c\hat{u}' = (\delta b)u'' - (\delta c)u', \\ \hat{u}(0) = 0,\ \hat{u}(1) = 0. \end{cases} \tag{4}$$

$\Rightarrow$ We want to be able to reformulate the directional derivative (3), in order to obtain a calculable expression for the gradient.

$\Rightarrow$ So we multiply the tangent linear model (4) by a variable p and we integrate twice by parts, transferring derivatives from $\hat{u}$ onto p:

$$-b\int_0^1 \hat{u}''p\,\mathrm{d}x + c\int_0^1 \hat{u}'p\,\mathrm{d}x$$
$$= \int_0^1 \left((\delta b)u''\,\mathrm{d}x - (\delta c)u'\right)p\,\mathrm{d}x,$$

which gives (term-by-term)

$$\begin{aligned}
\int_0^1 \hat{u}''p\,\mathrm{d}x &= [\hat{u}'p]_0^1 - \int_0^1 \hat{u}'p'\,\mathrm{d}x \\
&= [\hat{u}'p - \hat{u}p']_0^1 + \int_0^1 \hat{u}p''\,\mathrm{d}x \\
&= \hat{u}'(1)p(1) - \hat{u}'(0)p(0) + \int_0^1 \hat{u}p''\,\mathrm{d}x
\end{aligned}$$

and

$$
\begin{aligned}
\int_0^1 \hat{u}' p \,\mathrm{d}x &= [\hat{u}p]_0^1 - \int_0^1 \hat{u}p' \,\mathrm{d}x \\
&= -\int_0^1 \hat{u}p' \,\mathrm{d}x.
\end{aligned}
$$

$\Rightarrow$ Putting these results together, we have

$$
\begin{aligned}
-b\left(\hat{u}'(1)p(1) - \hat{u}'(0)p(0) + \int_0^1 \hat{u}p''\right)& \\
+c\left(-\int_0^1 \hat{u}p'\right)& \\
= \int_0^1 \left((\delta b)u'' - (\delta c)u'\right) p&
\end{aligned}
$$

or, grouping terms,

$$\int_0^1 \left(-bp'' - cp'\right)\hat{u} \tag{5}$$
$$= b\hat{u}'(1)p(1) - b\hat{u}'(0)p(0) \tag{6}$$
$$+ \int_0^1 \left((\delta b)u'' - (\delta c)u'\right)p. \tag{7}$$

⇒ Now, in order to get rid of *all* the terms in $\hat{u}$ in this expression, we impose that p must satisfy *the adjoint model*

$$\begin{cases} -bp'' - cp' = (u - u^{\mathrm{obs}}), \\ p(0) = 0, \; p(1) = 0. \end{cases} \tag{8}$$

⇒ Integrating (8) and using the expression (5), we obtain

$$
\begin{aligned}
\int_0^1 (u - u^{\mathrm{obs}})\hat{u} \\
= \int_0^1 \left(-bp'' - cp'\right)\hat{u} \\
= (\delta b)\left(\int_0^1 pu''\right) + (\delta c)\left(-\int_0^1 pu'\right).
\end{aligned}
$$

⇒ We recognize, in the last two terms, the L^2 inner product, which enables us, based on the definition of the directional derivative as the inner product between the gradient and the parameter perturbations,

$$\delta J \doteq \nabla_{\mathbf{m}} J \delta \mathbf{m},$$

to finally write an explicit expression for the gradient, based on (3),

$$\nabla J(b,c) = \left(\int_0^1 pu''\mathrm{d}x, -\int_0^1 pu'\mathrm{d}x\right)^{\mathrm{T}},$$

- or, separating the two components,

$$\begin{aligned} \nabla_b J(b,c) &= \int_0^1 p u'' \mathrm{d}x, && (9) \\ \nabla_c J(b,c) &= -\int_0^1 p u' \mathrm{d}x. && (10) \end{aligned}$$

- Thus, in this example, to compute the gradient of the least-squares error cost function, we must:

  ⇒ solve the direct equation (2) for u and derive u' and u'' from the solution, using some form of numerical differentiation (if we solved with finite differences), or differentiating the shape functions (if we solved with finite elements);

  ⇒ solve the adjoint equation (8) for p (using the same solver[3] that we used for u);

  ⇒ compute the two terms of the gradient, (9) and (10), using any suitable numerical integration scheme.

[3] This is not true when we use a *discrete* adjoint approach—see Section **??**

- **Conclusion**:

  ⇒ Thus for the additional cost of one solution of the adjoint model (8) plus a numerical integration, we can compute the gradient of the cost function with respect to either one or both of the unknown parameters.

  ⇒ It is now a relatively easy task to find (numerically) the optimal values of b and c that minimize J by a suitable descent algorithm, for example a quasi-Newton methods;

  ⇒ Note that this problem can be modified to treat the case where the observation is at one end-point, together with a discrete number of points in the interval.

    → In this case, the corresponding boundary condition must be of Neumann or mixed type, and the cost function needs to be modified, becoming a sum of the squared mismatches over the discrete observation points.

    → The adjoint boundary conditions and right-hand side will also be modified, but the expressions for the gradients will remain exactly the same.

Example: Lagrangian Adjoint Method for Parameter Identification

- We now consider a variant of the convection-diffusion example, where the diffusion coefficient is spatially varying.

  $\Rightarrow$ This model is closer to many physical situations, where the medium is not homogeneous and we have zones with differing diffusive properties.

- The system is now

$$\begin{cases} -\left(a(x)u'(x)\right)' - u'(x) = q(x), \quad 0 < x < 1, \\ u(0) = 0, \; u(1) = 0. \end{cases} \tag{11}$$

- with the cost function

$$J[a] = \frac{1}{2} \int_0^1 \left(u(x) - u^{\mathrm{obs}}(x)\right)^2 \, \mathrm{d}x,$$

where $u^{\mathrm{obs}}(x)$ denotes the observations on $[0,1]$.

- We now introduce an alternative approach for deriving the gradient, based on the Lagrangian (or variational formulation).

- Let the cost function

$$J^*[a,p] = \frac{1}{2}\int_0^1 \left(u(x) - u^{\mathrm{obs}}(x)\right)^2 \mathrm{d}x + \int_0^1 p\left(-\left(au'\right)' - u' - q\right)\mathrm{d}x,$$

noting that

$\Rightarrow$ the second integral is zero when u is a solution of (11)

$\Rightarrow$ and that the adjoint variable, p, can be considered here to be a Lagrange multiplier function.

- We begin by taking the variation of J^* with respect to its variables, a and p,

$$\begin{aligned}\delta J^* = & \int_0^1 \left(u - u^{\mathrm{obs}}\right) \delta u \,\mathrm{d}x \\ & + \int_0^1 \delta p \overbrace{\Big(-\left(a u'\right)' - u' - q\Big)}^{=0} \mathrm{d}x \\ & + \int_0^1 p \left[\left(-\delta a\, u' - a\,\delta u'\right)'\right].\end{aligned}$$

- Now the strategy is to "kill terms" by imposing suitable, well-chosen conditions on p.

$\Rightarrow$ This is achieved by integrating by parts and then defining the adjoint equation and boundary conditions on p as follows:

$$\begin{aligned}
\delta J^* &= \int_0^1 \left[(u - u^{\mathrm{obs}}) + p' - (ap')'\right] \delta u \,\mathrm{d}x \\
&\quad + \int_0^1 \delta a \, u' p' \mathrm{d}x \\
&\quad + \left[-p(\delta u + u'\delta a + a\delta u') + p' a \delta u\right]_0^1 \\
&= \int_0^1 \delta a \, u' p' \mathrm{d}x,
\end{aligned}$$

where

→ we have used the zero boundary conditions on δu

→ and assumed that the following adjoint system must be satisfied by p:

$$\begin{cases} -\,(ap')' + p' = -(u - u^{\mathrm{obs}}), \quad 0 < x < 1, \\ p(0) = 0, \; p(1) = 0. \end{cases} \tag{12}$$

⇒ And, as before, based on the key result relating the variation to the gradient, we are left with an explicit expression for the gradient,

$$\nabla_{a(x)} J^* = u' p'.$$

- Thus with

  $\Rightarrow$ one solution of the direct system (11) plus

  $\Rightarrow$ one solution of the adjoint system (12), we recover the gradient of the cost function with respect to the sought for diffusion coefficient, $a(x)$.

Example: Lagrangian Adjoint Method for Diffusion Equation

- The natural extension of the ordinary differential equations seen above is the initial-boundary-value problem known as the diffusion equation,

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

- This equation has multiple origins emanating from different physical situations.
  - ⇒ The most common application is particle diffusion, where u is a concentration and ν is a diffusion coefficient.
  - ⇒ Then there is heat diffusion, for which u is a temperature and ν is a thermal conductivity.
  - ⇒ The equation is also found in finance, being closely related to the Black-Scholes model.

⇒ Another important application is population dynamics.

⇒ These diverse application fields, and hence the diffusion equation, give rise to a number of inverse and data assimilation problems.

- A variety of different controls can be applied to this system:

⇒ *internal* control, $\nu(x)$: this is the parameter identification problem, also known as tomography;

⇒ *initial* control, $\xi(x) = u_0(x)$: this is a source detection IP or DA problem;

⇒ *boundary* control, $\eta(t) = u(L,t)$: this is the "classical" boundary control problem, also a parameter identification IP.

- As above, we can define the mismatch/L2 *cost function*,

$$J[\nu,\xi,\eta] = \frac{1}{LT}\int_0^T\int_0^L \left(u - u^{\mathrm{obs}}\right)^2 \,\mathrm{d}x\,\mathrm{d}t,$$

which is now a space-time multiple integral, and its related LAGRANGIAN,

$$J^* = \frac{1}{LT}\int_0^T\int_0^L \left(u - u^{\mathrm{obs}}\right)^2 \,\mathrm{d}x\,\mathrm{d}t + \frac{1}{LT}\int_0^T\int_0^L p\left[u_t - (\nu u_x)_x\right]\,\mathrm{d}x\,\mathrm{d}t.$$

- Now take the variation of J^*,

$$\begin{aligned}\delta J^* \quad = \quad & \frac{1}{LT}\int_0^T\int_0^L 2(u - u^{\mathrm{obs}})\delta u\,\mathrm{d}x\,\mathrm{d}t \\ & + \frac{1}{LT}\int_0^T\int_0^L \delta p\,\overbrace{\left[u_t - (\nu u_x)_x\right]}^{=0}\,\mathrm{d}x\,\mathrm{d}t \\ & + \frac{1}{LT}\int_0^T\int_0^L p\left[\delta u_t - (\delta\nu\, u_x + \nu\delta u_x)_x\right]\,\mathrm{d}x\,\mathrm{d}t,\end{aligned}$$

and perform integration by parts to obtain

$$\delta J^* = \frac{1}{LT}\int_0^T\int_0^L \delta\nu\, u_x p_x \,\mathrm{d}x\,\mathrm{d}t \tag{13}$$

$$- \frac{1}{LT}\int_0^L p\ \delta u|_{t=0}\,\mathrm{d}x \tag{14}$$

$$+ \frac{1}{LT}\int_0^T p\ \delta\eta|_{x=L}\,\mathrm{d}t, \tag{15}$$

where we have defined the adjoint equation as

$$\frac{\partial p}{\partial t} + \nabla\cdot(\nu\nabla u) = 2(u - u^{\mathrm{obs}}), \quad x\in(0,L), \quad t>0$$
$$p(0,t) = 0, \quad p(L,t) = 0,$$
$$p(x,T) = 0.$$

- As before, this equation is of the same type as the original diffusion equation, but must be solved backwards in time.
- Finally, from (13) we can pick off each of the three

desired terms of the gradient,

$$
\begin{aligned}
\nabla_{\nu(x)} J^* &= \frac{1}{T}\int_0^T u_x p_x \,\mathrm{d}t, \\
\nabla_{u|_{t=0}} J^* &= -\left. p\right|_{t=0}, \\
\nabla_{\eta|_{x=L}} J^* &= \left. p\right|_{x=L}.
\end{aligned}
$$

- **Conclusion**:
  - $\Rightarrow$ Once again, at the expense of a single (backward) solution of the adjoint equation, we obtain explicit expressions for the gradient of the cost function with respect to each of the three control variables. T
  - $\Rightarrow$ his is quite remarkable and completely avoids "brute force" or exhaustive minimization, though, as mentioned earlier, we only have the guarantee to find a local minimum.
  - $\Rightarrow$ However, if we have a good starting guess that is usually obtained from historical or other "phys-

ical'' knowledge of the system, we are sure to arrive at a good (or at least, better) minimum.

Adjoint Example (III) : Nonlinear PDE (Burgers)

- Burgers Equation (a simplified, but realistic model for Navier-Stokes) with control of the initial condition and boundary control.

- Viscous Burgers equation in the interval $x \in [0, L]$ is defined by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = f,$$
$$u(0,t) = \psi_1(t), \quad u(L,t) = \psi_2(t),$$
$$u(x,0) = u_0(x).$$

- The control vector

$$(u_0(x), \psi_1(t), \psi_2(t)).$$

- The cost function is taken as

$$J(u_0, \psi_1, \psi_2) = \frac{1}{2} \int_0^T \int_0^L \left(u - u^{\mathrm{obs}}\right)^2 \, dx \, dt.$$

- The adjoint model is

$$\begin{gathered} \frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} - \nu \frac{\partial^2 p}{\partial x^2} = u - u^{\mathrm{obs}}, \\ p(0, t) = 0, \quad p(L, t) = 0, \\ p(x, T) = 0. \end{gathered}$$

- Gradient/variation/directional derivative of J is finally,

$$\begin{aligned} \hat{J}\left[u_0, \psi_1, \psi_2\right](\delta_u, \delta_1, \delta_2) = & - \int_0^L \delta_u p(x, 0) \, dx \\ & + \int_0^T \nu \delta_2 \frac{\partial p}{\partial x}(L, t) \\ & - \nu \delta_1 \frac{\partial p}{\partial x}(0, t) \, dt \end{aligned}$$

that gives,

$$
\begin{aligned}
\nabla_{u_0} J &= -p(x, t=0) \\
\nabla_{\psi_1} J &= -\nu \frac{\partial p}{\partial x}(x=0, t) \\
\nabla_{\psi_2} J &= \nu \frac{\partial p}{\partial x}(x=L, t).
\end{aligned}
$$

- These explicit gradients enable us to solve inverse problems for

  $\Rightarrow$ the initial condition, which is a data assimilation problem;

  $\Rightarrow$ or for the boundary conditions, which is an optimal boundary control problem;

  $\Rightarrow$ or for both.

- Another extension would be a parameter identification problem for the viscosity ν. This would make an excellent project or advanced exercise.

Codes

Various open-source repositories and codes are available for both academic and operational data assimilation.

1. DARC: `https://research.reading.ac.uk/met-darc/` from Reading, UK.

2. DAPPER: `https://github.com/nansencenter/DAPPER` from Nansen, Norway.

3. DART: `https://dart.ucar.edu/` from NCAR, US, specialized in ensemble DA.

4. OpenDA: `https://www.openda.org/`.

5. Verdandi: `http://verdandi.sourceforge.net/` from INRIA, France.

6. PyDA: `https://github.com/Shady-Ahmed/PyDA`, a Python implementation for academic use.

7. Filterpy: `https://github.com/rlabbe/filterpy`, dedicated to KF variants.

8. EnKF; `https://enkf.nersc.no/`, the original Ensemble KF from Geir Evensen.

References

1. J. Nocedal, S.J. Wright. *Numerical Optimization*. Springer, 2006.

2. F. Tröltzsch. *Optimal Control of Partial Differential Equations*. AMS, 2010.

3. K. Law, A. Stuart, K. Zygalakis. *Data Assimilation. A Mathematical Introduction*. Springer, 2015.

4. G. Evensen. *Data assimilation, The Ensemble Kalman Filter*, 2nd ed., Springer, 2009.

5. A. Tarantola. *Inverse problem theory and methods for model parameter estimation.* SIAM. 2005.

6. O. Talagrand. Assimilation of observations, an introduction. *J. Meteorological Soc. Japan*, **75**, 191–209, 1997.

7. F.X. Le Dimet, O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus,* **38**(2), 97–110, 1986.

8. J.-L. Lions. Exact controllability, stabilization and perturbations for distributed systems. *SIAM Rev.*, **30**(1):1–68, 1988.