

# Differentiating functionals

## Finite differencing

The definition of the derivative  $d\hat{J}/dm$  is

$$\frac{d\hat{J}}{dm_i} = \lim_{h \rightarrow 0} \frac{\hat{J}(m + he_i) - \hat{J}(m)}{h}$$

where  $e_i$  is the vector with 0 in all entries except for 1 in the  $i^{\text{th}}$  entry. Each component of the gradient vector  $d\hat{J}/dm$  is the derivative of the functional  $\hat{J}$  with respect to the corresponding component of  $m$ . A simple idea for approximating the derivative is to compute each component of the gradient as

$$\frac{d\hat{J}}{dm_i} \approx \frac{\hat{J}(m + he_i) - \hat{J}(m)}{h}$$

for some small choice of  $h$ . The advantage of this approach is that it is very straightforward: it still only requires a black-box evaluator for  $\hat{J}$ , and the approximation of the gradients can be done entirely within the optimisation algorithm.

However, this approach suffers from several serious drawbacks. One problem is that it is not obvious how to choose an appropriate value for  $h$ : choose  $h$  too large, and the finite difference will not approximate the limit value; choose  $h$  too small, and numerical precision will destroy the accuracy of the approximation. A more serious problem, however, is that this approximation requires one functional evaluation for each degree of freedom in the parameter space. When each functional evaluation requires an expensive PDE solve, this approach quickly becomes impractical, and a more intelligent algorithm is required.

### When finite differencing is useful

In the PDE-constrained optimisation case, finite differencing isn't very useful for computing the gradient of the functional. However, it is *very* useful for rigorously *verifying* gradients computed with another approach. For more details, see the section of the dolfin-adjoint documentation on [verifying functional gradients](#).

## The tangent linear approach

Recall that  $\hat{J}(m)$  is the functional considered as a pure function of  $m$ :

$$\hat{J}(m) = J(u(m), m).$$

Let us apply the chain rule to  $\hat{J}(m)$ :

$$\frac{d\hat{J}}{dm} = \frac{\partial J}{\partial u} \frac{du}{dm} + \frac{\partial J}{\partial m}.$$

$\begin{matrix} 1 \times M & 1 \times U & U \times M & 1 \times M \end{matrix}$

Let us inspect each term of this relationship, and build up some intuition about each.  $\partial J/\partial m$  and  $\partial J/\partial u$  are typically very straightforward to compute:  $J$  is usually a simple closed-form expression in terms of  $u$  and  $m$ , and so their differentiation by hand is generally trivial. Both of these quantities are vectors, with dimensions of the parameter space and solution space respectively. By contrast, the solution Jacobian  $du/dm$  is rather difficult to compute. This object is a massive dense matrix, of dimensions (solution space  $\times$  parameter space), and as such it is unlikely to fit in memory. However, let us temporarily suppose that the number of parameters is small, and that we would like to compute  $d\hat{J}/dm$  using the relationship above.

With the PDE  $F(u, m) = 0$ , we have an relationship for  $u$  as an implicit function of  $m$ . If we take the total derivative of this equation with respect to  $m$ , we will have a relationship for the solution Jacobian  $du/dm$ :

$$\begin{aligned} \frac{d}{dm} F(u, m) &= \frac{d}{dm} 0 \\ \Rightarrow \frac{\partial F(u, m)}{\partial u} \frac{du}{dm} + \frac{\partial F(u, m)}{\partial m} &= 0 \\ \Rightarrow \frac{\partial F(u, m)}{\partial u} \frac{du}{dm} &= - \frac{\partial F(u, m)}{\partial m}. \end{aligned}$$

$\begin{matrix} U \times U & U \times M & U \times M \end{matrix}$

 v: release ▾

This last relationship is **the tangent linear equation** (or tangent linear system) associated with the PDE  $F(u, m) = 0$ . Let us carefully consider each term in the tangent linear system, and build up some intuition about each.

$du/dm$  is the solution Jacobian again, with which we can compute the functional gradient  $d\hat{J}/dm$ . It is the prognostic variable of this equation, the unknown quantity in the tangent linear system.

Now consider  $\partial F(u, m)/\partial u$ . Since  $F$  is a vector expression, its derivative with respect to  $u$  is an operator (a matrix); this operator acts on the solution Jacobian, **and therefore must be inverted or solved**.  $F(u, m)$  may have been nonlinear in  $u$ , but  $\partial F(u, m)/\partial u$  is always linear. In other words,  $\partial F(u, m)/\partial u$  is the linearisation of the equation operator, linearised about a particular solution  $u$ . If  $F(u, m)$  happened to be linear in the first place, and so  $F(u, m) \equiv A(m)u - b(m)$  for some operator  $A(m)$ , then  $\partial F(u, m)/\partial u$  is just the operator  $A(m)$  back again.

Finally, consider the term  $\partial F(u, m)/\partial m$ . Like  $du/dm$ , this is a matrix of dimension (solution space  $\times$  parameter space). This term acts as the source term for the tangent linear system; each column of  $\partial F(u, m)/\partial m$  provides the source term for the derivative of  $u$  with respect to one scalar entry in the parameter vector.

So, *when is solving the tangent linear system a sensible approach?* To answer this question, notice that we had to specify some parameter  $m$  to construct the tangent linear system, but that the functional  $J$  does not appear at all. In other words, **for a given parameter (input), the tangent linear solution can be used to easily compute the gradient of any functional**. This means that solving the tangent linear system makes sense *when there are a small number of parameters (inputs), and a large number of functionals of interest (outputs)*. However, this is generally not the case in PDE-constrained optimisation. Is there a better way?

## The adjoint approach

Let us rephrase the tangent linear approach to computing the gradient. We start by fixing our choice of parameter  $m$ , and then solve for the solution Jacobian  $du/dm$  associated with that choice of  $m$ . With this quantity in hand, we take its inner product with a source term  $\partial J/\partial u$  particular to the functional  $J$ , and can then compute the gradient  $d\hat{J}/dm$ .

Notice that we first fixed the parameter  $m$ , (the “denominator” of the gradient  $d\hat{J}/dm$ ) and *then* chose which functional we wished to compute the gradient of (the “numerator” of the gradient). Is there a way where we could do the opposite: first fix the functional  $J$ , and *then* choose which parameter to take the gradient with respect to? The answer is yes, and that approach is referred to as the adjoint approach.

Suppose the tangent linear system is invertible. Then we can rewrite the solution Jacobian as

$$\frac{du}{dm} = - \left( \frac{\partial F(u, m)}{\partial u} \right)^{-1} \frac{\partial F(u, m)}{\partial m}.$$

We usually could not compute this expression (computing the inverse of the operator  $\partial F(u, m)/\partial u$  is prohibitive), but we can still use it and reason about it. Let us substitute this expression for the solution Jacobian into the expression for the gradient of  $\hat{J}$ :

只是倒置, -1不可去掉的, 倒置与求逆不要混淆

$$\begin{aligned} \frac{d\hat{J}}{dm} &= \frac{\partial J}{\partial u} \frac{du}{dm} + \frac{\partial J}{\partial m} \\ \Rightarrow \frac{d\hat{J}}{dm} &= - \frac{\partial J}{\partial u} \left( \frac{\partial F(u, m)}{\partial u} \right)^{-1} \frac{\partial F(u, m)}{\partial m} + \frac{\partial J}{\partial m}. \end{aligned}$$

Now let's take the adjoint (Hermitian transpose) of the above equation:

$$\frac{d\hat{J}^*}{dm^*} = - \frac{\partial F^*}{\partial m^*} \frac{\partial F^*}{\partial u^*} \frac{\partial J^*}{\partial u^*} + \frac{\partial J^*}{\partial m^*}$$

Let us gather the solution of the inverse Jacobian acting on a vector, and define it to be a new variable:

### The tangent linear system

The tangent linear system is the same idea as the *forward mode* of algorithmic or automatic differentiation.

### The adjoint of a matrix

The notation  $A^*$  means to take the transpose of  $A$ ,  $A^T$ , and take the complex conjugate of each entry. **If the matrix  $A$  is composed entirely of real numbers, then the adjoint is just the transpose**. Other words for the adjoint are the Hermitian and the conjugate transpose.

$$\lambda = \left( \frac{\partial F(u, m)}{\partial u} \right)^{-*} \frac{\partial J^*}{\partial u}$$

$$\Rightarrow \left( \frac{\partial F(u, m)}{\partial u} \right)^* \lambda = \frac{\partial J^*}{\partial u}.$$

This relationship is the **adjoint equation** (or adjoint system) associated with the PDE  $F(u, m) = 0$ . Again, let us carefully consider each term in the adjoint equation and build up some intuition about each.

$\lambda$  is the *adjoint variable associated with  $u$* . Each component of the solution  $u$  will have a corresponding adjoint variable. For example, if  $F$  is the Navier-Stokes equations, and  $u$  is the tuple of velocity and pressure, then  $\lambda$  is the tuple of adjoint velocity and adjoint pressure. Similarly, if the problem is time-dependent, the adjoint is also time-dependent, with each variable through time having a corresponding adjoint value.

$(\partial F(u, m)/\partial u)^*$  is the *adjoint of the tangent linear operator*. Commonly, this is referred as the “adjoint operator”. By taking the transpose, we *reverse the flow of information in the equation system*. For example, if a tracer is advected downstream (and so information about upstream conditions is advected with it), the adjoint PDE advects information in the reverse sense, i.e. upstream. This extends to the temporal propagation of information: if  $F(u, m)$  is a time-dependent PDE (and so propagates information from earlier times to later times), the adjoint PDE *runs backwards in time* (propagates information from later times to earlier times). This property will be examined in more detail in the next section.

$\partial J/\partial u$  is the source term for the adjoint equation. It is this source term that makes an adjoint solution *specific to a particular functional*. Just as one cannot speak of the tangent linear solution without referring to a particular choice of parameter, one cannot speak of the adjoint solution without referring to a specific choice of functional.

As the tangent linear operator is always linear, the adjoint is linear in  $u$  also, and so the adjoint equation is always linear in  $\lambda$ . This property will also be examined in more detail in the next section.



So, to compute the functional gradient  $d\hat{J}/dm$ , we **first solve the adjoint equation for  $\lambda$**  (fixing the “nominator” of the gradient, as the adjoint is specific to the functional), and then take its inner product with respect to  $-\partial F(u, m)/\partial m$  to compute the gradient with respect to a particular parameter  $m$  (fixing the “denominator” of the gradient). This is precisely the *dual* approach to that of computing  $d\hat{J}/dm$  using the tangent linear approach, and has precisely the dual scaling: **for a given functional (output), the adjoint solution can be used to easily compute the gradient with respect to any parameter**. Therefore, solving the adjoint system is extremely efficient *when there are a small number of functionals (outputs), and a large number of parameters (inputs)*. This is precisely the case we are considering in PDE-constrained optimisation: there is one functional (output) of interest, but many parameters.

So, with some knowledge of the chain rule and some transposition, we have devised an algorithm for computing the gradient  $d\hat{J}/dm$  that is extremely efficient for our case where we have many parameters and only one functional.

## Summary

A sketch of the solution approach for the PDE-constrained optimisation problem is therefore:

1. Start with some initial guess for the parameters  $m$ .
2. Compute the functional  $\hat{J}(m)$  (using the forward model) and its gradient (using the adjoint model).
3. Pass these values to an optimisation algorithm. This algorithm returns a new point in parameter space with a better functional value.
4. If the gradient is zero, or if the maximum number of iterations has been reached, terminate. Otherwise, go to step 2.

Of course, PDE-constrained optimisation is a much richer field than the simple sketch above would suggest. Much work is focussed on exploiting particular properties of the equations or the functional, ensuring the gradient is represented with the correct Riesz representer, or imposing additional constraints on the parameter space, or exploiting advanced forward modelling concepts such as error estimation, goal-based adaptivity and reduced-order modelling. Nevertheless, although complications proliferate, the above algorithm captures the key idea of many approaches  [v: release](#)  for solving problems of enormous importance.

### The adjoint system

The adjoint system is the same idea as the *reverse mode* of algorithmic or automatic differentiation.

Another word for “adjoint” used in the literature is “dual”: people refer to the dual system, the dual solution, etc.

With the adjoint and tangent linear equations now introduced, let us examine them more thoroughly, in [the next section](#).

## References