Notes on adjoints

November 9, 2023

A linear operator may be represented as a matrix operation, $\mathbf{A}$ (having an input $\mathbf{u}$, and an output, $\mathbf{v}$):

$$\mathbf{v} = \mathbf{A}\mathbf{u}. \tag{1}$$

If $\mathbf{u}$ has n_u elements and $\mathbf{v}$ has n_v elements, then $\mathbf{A}$ is an $n_v \times n_u$ matrix. This is shorthand for the following element-by-element operations:

$$\mathbf{v}_i = \sum_{j=1}^{n_u} \mathbf{A}_{ij}\mathbf{u}_j. \tag{2}$$

Thus element $\mathbf{A}_{ij}$ is the coefficient that contributes to element i in the output vector $\mathbf{v}$ due to element j in the input vector $\mathbf{u}$. This is sometimes called the forward operator.

The transpose of this linear operation may be represented as the following matrix operation:

$$\hat{\mathbf{u}} = \mathbf{A}^\intercal \hat{\mathbf{v}}, \tag{3}$$

where $\hat{\mathbf{u}}$ has n_u elements, $\hat{\mathbf{v}}$ has n_v elements, and $\mathbf{A}^\intercal$ is the $n_u \times n_v$ matrix found by transposing $\mathbf{A}$ (rows become columns, and vice-versa). This is shorthand for the following:

$$\hat{\mathbf{u}}_j = \sum_{i=1}^{n_v} \mathbf{A}^\intercal_{ji}\hat{\mathbf{v}}_i. \tag{4}$$

Thus element $\mathbf{A}^\intercal_{ji}$ is the coefficient that contributes to element j in the output vector $\hat{\mathbf{u}}$ due to element i in the input vector $\hat{\mathbf{v}}$. Let us assume here that an adjoint is the same thing as a transpose.

This method of producing an adjoint is straightforward when the linear operations are represented as matrices. Many linear operators are represented as lines of code in a subroutine though, so we need a procedure to produce the adjoint version of any linear operator.

Note that $\mathbf{A}^\intercal_{ji} = \mathbf{A}_{ij}$, which makes (4) into

$$\hat{\mathbf{u}}_j = \sum_{i=1}^{n_v} \mathbf{A}_{ij}\hat{\mathbf{v}}_i. \tag{5}$$

What does this tell us about how to produce the adjoint version of lines of forward code? Let us compare (2) and (5). Suppose that in the forward code we have an operation that is something like

```
v(i) = f * u(j)
```

The fact that $\mathbf{v}_i$ is being updated using information from $\mathbf{u}_j$ means that, from (2), the factor f is effectively matrix element $\mathbf{A}_{ij}$. This is exactly the same matrix element that appears in (5), which means that in the adjoint code, the same factor, f, is used to update $\hat{\mathbf{u}}_j$ using information from $\hat{\mathbf{v}}_i$:

```
u_hat(j) = u_hat(j) + f * v_hat(i)
```

Notice that the output component is being incremented instead of being just set. This is done because this operation may not be the only one that contributes to $\hat{\mathbf{u}}_j$. For instance, there may be another line in the forward code

```
v(i1) = g * u(j)
```

This is equivalent to matrix element $\mathbf{A}_{i_1 j}$. In the adjoint, this will also update element j:

```
u_hat(j) = u_hat(j) + g * v_hat(i1)
```

Since this is an incremental form, the output vector is normally inititialised to zero at the start (unless the particular routine represents just part of the operator $\mathbf{A}$ or $\mathbf{A}^\intercal$, in which case the initialisation is done outside of the particular adjoint subroutine). This incremental form should be used unless it is clear that a step like

```
v(i) = f * u(j)
```

is the only step in the forward code that depends on the particular element $\mathbf{u}_j$.

Once the forward and adjoint routines have been coded (e.g. as follows):

```
subroutine A (u, v)
double precision, intent(in)  :: u(:)
double precision, intent(out) :: v(:)
...
end subroutine A


subroutine A_adj (v_hat, u_hat)
double precision, intent(in)  :: v_hat(:)
double precision, intent(out) :: u_hat(:)
...
end subroutine A_adj
```

it should be checked that the adjoint is correctly coded. This can be done with the adjoint test as follows

$$\text{does } (\mathbf{A}\mathbf{u})^{\intercal}(\mathbf{A}\mathbf{u}) = \mathbf{u}^{\intercal}\mathbf{A}^{\intercal}\mathbf{A}\mathbf{u}? \tag{6}$$

This test is performed by the following procedure with respect to the left hand side (LHS) and right hand side (RHS) of (6).

1. Set $\mathbf{u}$ to some random numbers.
2. Let $\mathbf{v} = \mathbf{A}\mathbf{u}$ by calling the forward subroutine.
3. Let $\hat{\mathbf{u}} = \mathbf{A}^{\intercal}\mathbf{v}$ by calling the adjoint subroutine.
4. Calculate the LHS: LHS $= \mathbf{v}^{\intercal}\mathbf{v}$.
5. Calculate the RHS: RHS $= \mathbf{u}^{\intercal}\hat{\mathbf{u}}$.
6. If LHS = RHS <u>to machine precision</u>, then the adjoint test is passed.