Page 24, Software update, gdapy01_07, dot product

I am rather aghast that the new versions of numpy, scipy and matplotlib have depreciated some common coding "phrases". The one relevant here is that a 1×1 array no longer count as a scalar. This breaks a lot of the code. So I am now (May 24, 2024) coding the dot product $s = \mathbf{a} \cdot \mathbf{b}$ as

```
s=np.matmul(a.T,b); s=s[0,0];
```

and not as

```
s=np.matmul(a.T,b);
```

That is, `s` is overwritten by its first element. In cases where the dot product appears in an expression, I have broken it out into several lines. E.g. what was

```
t=a/sqrt(np.matmul(a.T,a));
```

becomes

```
asq=np.matmul(a.T,a); asq=asq[0,0];
t=a/sqrt(asq);
```

I have not (yet) made any effort to add to these errata specific places in the book where this kind of change has been made.

Page 86, Software update, gdapy04_09, `las.bicg()`

I am rather aghast that the new versions of numpy, scipy and matplotlib have depreciated some common coding methods. The one relevant here is that the `tol` keyword of `la.bicg()` has been changed to `rtol`. This breaks a lot of the code.

Thus the new command

```
q=la.bicg(LO,FTh,rtol=tol,maxiter=maxit);
```

replaces the old

```
q=la.bicg(LO,FTh,tol=tol,maxiter=maxit);
```

Also, I am now explicitly casting the result to a `float`, with the new command

```
mest=gda_cvec(q[0].astype(float));
```

which replaces

```
mest=gda_cvec(q[0]);
```

However, I'm not sure this is necessary. But it doesn't hurt.

I have not (yet) made any effort to add to these errata specific places in the book where this kind of change has been made.

Page 143, Equation 7.32, correction

Delete factor of Δt as shown

$$\frac{\mathbf{m}^{(m)} - \mathbf{m}^{(m-1)}}{\Delta t} = \mathbf{Z}\,\mathbf{m}^{(m-1)} + \Delta t\,\mathbf{q}^{(m-1)} \tag{7.32}$$

Page 143, Equation 7.33, correction

Change of n to m as shown

$$\mathbf{m}^{(m)} = \mathbf{D}\mathbf{m}^{(m-1)} + \mathbf{s}^{(m-1)} \quad \text{or} \quad -\mathbf{D}\mathbf{m}^{(m-1)} + \mathbf{I}\mathbf{m}^{(m)} = \mathbf{s}^{(m-1)}$$
$$\text{with } \mathbf{D} \equiv \Delta t\mathbf{Z} + \mathbf{I} \text{ and } \mathbf{s}^{(m)} \equiv \Delta t\mathbf{q}^{(n)} \tag{7.33}$$

m

Page 149, Problem 7.2, clarification

Change the word “car” to “object” and insert the word “position” as indicted.

object

7.2 An object moving horizontally in the u-direction is decelerating due to a resistive force $f = -cmv$ that is proportional to its mass m, velocity v, and a constant $c = 0.01$. The ~~car~~ starts at position $u = 0$ with initial velocity $v_0 = 1$. Newton's law implies $du/dt = v$ and $dv/dt = -cv$. (A) What are the dynamical quantities $\mathbf{Z}$, $\mathbf{q}$, $\mathbf{D}$, $\mathbf{s}$ in Eqs. (7.32) and (7.33)? (B) Create noise-free data by stepping Eq. (7.33) forward in time (say, with $\Delta t = 1$). (C) Create noisy data by adding Normally distributed random noise with zero mean and variance $\sigma_d^2 = (0.1)^2$ to the noise-free data. (D) Assuming that the dynamical equation (which is ultra-simplified) has variance $\sigma_s^2 = (0.01)^2$, use Kalman filtering to estimate $u(t)$. Comment upon the results.

position

Page 208, Section 11.6, addendum

Add the following sentence at the end of Section 11.6:

The posterior covariance of the solution can be estimated as:

$$\left[\operatorname{cov}\mathbf{x}^{(n)}\right] \approx (\mathbf{I} - [\operatorname{cov}\mathbf{x}]\mathbf{F}^T[\mathbf{F}[\operatorname{cov}\mathbf{x}]\mathbf{F}^T]^{-1}\mathbf{F})[\operatorname{cov}\mathbf{x}]$$

where $\mathbf{F} \equiv \mathbf{F}^{(p)}$ is from the last iteration.