

## 190.5

THOMAS BREYDO

**Problem.** On  $\mathcal{P}_2(\mathbf{R})$ , consider the inner product given by

$$\langle p, q \rangle = \int_0^1 p(x)q(x) \, dx.$$

Apply the Gram–Schmidt Procedure to the basis  $1, x, x^2$  to produce an orthonormal basis of  $\mathcal{P}_2(\mathbf{R})$ .

**Claim.** We can write code that applies the Procedure.

*Proof.* Proof by construction, voila:

```
1 def gram_schmidt(vectors: list[Vector]) -> list[Vector]:
2     output = []
3     for v in vectors:
4         next_e = sum([-e.scaled(v * e) for e in output], start=v)
5         output.append(next_e.normalized())
6     return output
```

□

**Note.** If you're curious about how I implemented the `Vector` objects so that they can be scaled, multiplied, etc, check out my code [here](#).

**Claim.** We can write a function that computes

$$\int_0^1 p(x)q(x) \, dx.$$

*Proof.* Suppose

$$\begin{aligned} p(x) &= a_0 + a_1x + \cdots + a_nx^n \\ q(x) &= b_0 + b_1x + \cdots + b_nx^n. \end{aligned}$$

Then,

$$\begin{aligned}
 \int_0^1 p(x)q(x) \, dx &= \int_0^1 (a_0 + a_1x + \cdots + a_nx^n)(b_0 + b_1x + \cdots + b_nx^n) \, dx \\
 &= \int_0^1 a_0b_0 \, dx + \int_0^1 a_0b_1x \, dx + \cdots + \int_0^1 a_0b_nx^n \, dx \\
 &\quad + \int_0^1 a_1b_0 \, dx + \int_0^1 a_1b_1x \, dx + \cdots + \int_0^1 a_1b_nx^n \, dx \\
 &\quad \vdots \\
 &\quad + \int_0^1 a_nb_0 \, dx + \int_0^1 a_nb_1x \, dx + \cdots + \int_0^1 a_nb_nx^n \, dx.
 \end{aligned}$$

Each monomial is easy to integrate, since

$$\begin{aligned}
 \int_0^1 a_ix^ib_jx^j \, dx &= a_ib_j \int_0^1 x^{i+j} \, dx \\
 &= \frac{a_ib_j}{i+j+1} x^{i+j+1} \Big|_0^1 \\
 &= \frac{a_ib_j}{i+j+1}.
 \end{aligned}$$

You can see this fact used on line 7 below.

```

1 def inner_product(p, q):
2     """Evaluate the definite integral from 0 to 1 of pq."""
3     total = 0
4     for exp1, coef1 in enumerate(p.components):
5         for exp2, coef2 in enumerate(q.components):
6             # integral of (coef1)x^exp1 + (coef2)x^exp2 is:
7             total += coef1 * coef2 / (exp1 + exp2 + 1)
8     return total

```

□

**Claim.** The orthogonal basis is

$$\begin{aligned}
 e_1 &= 1 \\
 e_2 &\approx -1.73 + 3.456x \\
 e_3 &\approx 2.23 - 13.42x + 13.42x^2.
 \end{aligned}$$

*Proof.* The following snippet of code produces the desired result. You can confirm the orthogonality of these polynomials by looking at [this](#) Desmos project.

```

1 basis = [BasisVector("1"), BasisVector("x"), BasisVector("x^2")]
2 V = VectorSpace(basis, inner_product)
3 u = Vector([1, 0, 0], space=V)
4 v = Vector([0, 1, 0], space=V)
5 w = Vector([0, 0, 1], space=V)
6
7 for vec in gram_schmidt([u, v, w]):
8     print(vec)

```

190.5

3



**Note.** You can view the source code for this solution [here](#).