PHYS 301

HOMEWORK #5

Due : 20 Feb. 2013--Solutions

1. What is the value of the product $\epsilon_{ijk}\,\epsilon_{ijk}$ where ϵ is the Levi-Civita permutation tensor?

Solution : A formal solution starts by noting there are 3 repeated indices (i, j, and k), so the product represents the triple sum :

$$\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3} \epsilon_{ijk}\,\epsilon_{ijk} =$$

$$\epsilon_{111}\,\epsilon_{111} + \epsilon_{112}\,\epsilon_{112} + \epsilon_{113}\,\epsilon_{113} + \epsilon_{121}\,\epsilon_{121} + \epsilon_{122}\,\epsilon_{122} + \epsilon_{123}\,\epsilon_{123} + \epsilon_{211}\,\epsilon_{211} \ldots + 20 \text{ further terms}$$

The triple sum will produce 27 individual terms, each of which is a product. The value of 21 of the products will be 0 x 0, but 3 of them will be (1) (1) and another three will be (-1) (-1), so the value of the triple sum is + 6.

---

2. Use summation notation to prove that $\nabla(f\,g) = f\,\nabla g + g\,\nabla f$ where f and g are scalar functions.

Solution : We write the left hand side in summation notation :

$$\nabla(f\,g) \;\rightarrow\; \frac{\partial}{\partial x_i}(f\,g)$$

Applying the product rule to f g :

$$\frac{\partial}{\partial x_i}(f\,g) \;=\; f\,\frac{\partial}{\partial x_i}g \;+\; g\,\frac{\partial}{\partial x_i}f \;=\; f\,\nabla g \;+\; g\,\nabla f$$

---

3. Consider the function f (x) = x^3 on (-1, 1); find the Fourier coefficients for this function and then use Parseval' s theorem to evaluate :

$$\sum_{n=1}^{\infty} \frac{1}{n^6}$$

You may (and should) use results from previous homework problems. You may use Mathematica to determine the expressions for the Fourier coefficients (including evaluating definite integrals and refining those results for integer values of n).

Solution :

Parseval' s Theorem relates the average value of the square of a function to the Fourier coefficients :

$$\text{average value of } (f(x))^2 = \left(\frac{a_0}{2}\right)^2 + \frac{1}{2}\sum_{n=1}^{\infty} a_n^2 + \frac{1}{2}\sum_{n=1}^{\infty} b_n^2$$

For the given function on (-1, 1), the average value of $(x^3)^2$ is:

$$\text{average value of } x^6 = \frac{\int_{-1}^{1}(x^3)^2\, dx}{\text{length of interval}} = \frac{1}{2}\int_{-1}^{1} x^6\, dx = \frac{1}{7}$$

Because f (x) is odd, we can use symmetry arguments to set all the a coefficients to zero; we find the values of the b coefficients from :

$$b_n = \frac{1}{1}\int_{-1}^{1} x^3 \sin(n\pi x/1)\, dx = -2(-1)^n \frac{(-6 + n^2\pi^2)}{n^3\pi^3}$$

$$b_n^2 = 4\frac{(36 - 12 n^2\pi^2 + n^4\pi^4)}{n^6\pi^6} = \frac{144}{n^6\pi^6} - \frac{48}{n^4\pi^4} + \frac{4}{n^2\pi^2}$$

Substituting these results into Parseval' s Theorem, we get :

$$\frac{1}{7} = \frac{1}{2}\left(\sum_{n=1}^{\infty}\frac{144}{n^6\pi^6} - \sum_{n=1}^{\infty}\frac{48}{n^4\pi^4} + \sum_{n=1}^{\infty}\frac{4}{n^2\pi^2}\right) \tag{1}$$

We know from the last homework, that :

$$\sum_{n=1}^{\infty}\frac{1}{n^2} = \frac{\pi^2}{6} \quad \text{and} \quad \sum_{n=1}^{\infty}\frac{1}{n^4} = \frac{\pi^4}{90}$$

Substituting these values into eq. (1), we get :

$$\frac{1}{7} = 72\sum_{n=1}^{\infty}\frac{1}{n^6\pi^6} - \frac{24}{90} + \frac{1}{3} \Rightarrow \frac{8}{105} = 72\sum_{n=1}^{\infty}\frac{1}{n^6\pi^6} \Rightarrow \sum_{n=1}^{\infty}\frac{1}{n^6} = \frac{\pi^6}{945}$$

---

4. In this problem, you will write a short Mathematica program to estimate the square root of a number using Newton' s Method. Your solution should make use of recursive relations and loop controls that we have studied in lab. Use the RandomInteger function in Mathematica to generate a number between 1, 000, 000 and 10, 000, 000; this will be the number whose square root you will find. Your initial estimate should be 1, and you will iterate until the nth esimate differs from the (n - 1) st estimate by less than 0.001. Your output should show explicitly and clearly: a) the initial number, b) its square root, and c) how many iterations it took to produce that answer. Do not do any direct calculation of the square root using Sqrt or N^1/2 or similar functions. This question will be worth 30 points (all other questions on this homework assignment are worth 10 points each.)

Solutions :

Using a while statement :

In[149]:=
```
Clear[f, number]
number = RandomInteger[{1 000 000, 10 000 000}];
f[1] = 1;
f[n_] := 0.5 (number / f[n - 1] + f[n - 1])
n = 2; While[Abs[f[n] - f[n - 1]] > 0.001, n++]
Print["The square root of ", number, " = ",
 f[n] ".  The method required ", n, " iterations to converge."]
```

```
The square root of 3 860 965 = 1964.93 .  The method required 16 iterations to converge.
```

In this program, the first line clears any previous values of the variable number and the function f. The second line uses the random integer generator to find the number whose square root we will find; the third line sets our initial estimate of the square root to 1.

The fourth defines the new value of f in terms of the previous value of f. In the fifth line, do you see why we set n = 2?

The While statement uses as its test Abs[f[n] - f[n - 1]] > 0.001; this tests to see if the absolute value of the difference between the nth and (n -1) st estimate differ by more than 0.001. If they do, the process continues and sets n -> n + 1. If the test fails (i.e., the new value differs from the old value by less than 0.001), the program exits the loop and outputs the most recent values of n and f (n).

Using an essentially equivalent program but with a For statement :

In[209]:=
```
Clear[f, number]
number = RandomInteger[{1 000 000, 10 000 000}];
f[1] = 1;
f[i_] := 0.5 (number / f[i - 1] + f[i - 1])
For[i = 2, Abs[f[i] - f[i - 1]] > 0.001, i++]
Print["The square root of ", number, " is ", f[i],
 ".  The algorithm required ", i, " iterations to converge."]
```

```
The square root of 9 639 825 is 3104.81.  The algorithm required 17 iterations to converge.
```

Throw and Catch: We can use this problem to introduce a very useful Mathemtica tool, Catch/Throw statements. We can write this program using a Do loop and Catch Throw statements to exit the loop once the condition that
| f (n) - f (n - 1) | < 0.001 is met.

In[170]:=
```
Clear[number, f]
number = RandomInteger[{1 000 000, 10 000 000}];
f[1] = 1; f[n_] := 0.5 (number / f[n - 1] + f[n - 1])
Catch[Do[If[Abs[f[n] - f[n - 1]] < 0.001, Throw[Print["The square root of ", number, " = ",
     f[n] ".  The method required ", n, " iterations to converge."]]], {n, 2, 100}]]
```

```
The square root of 1 258 570 = 1121.86 .  The method required 16 iterations to converge.
```

I choose some arbitrarily large enough number for the upper limit of the do loop (in this case 100); I know from prior experience that this technique will take only 16 or 17 iterations to converge, so by setting the do loop to 100 iterations, I know we will achieve convergence long before the do loop exits on its own. The condition on this If statement is met if the absolute difference between successive iterations is less than 0.001; when this condition is true, the Throw statement exits the do loop. The Catch statement "Catches" the output of the do loop and outputs it. Throw statements are a good way to exit a loop when a condition is met; but every Throw statement must have an accompanying Catch statement.