Mathematical Experiments for Mathematics Majors

Michael Monagan
Simon Fraser University

Computer Algebra in Education
ACA 2025

MACM 204-2 Computing with Calculus using Maple

Prerequisites: Integral Calculus and Programming 1.

Goals

1. Practice/review programming skills on mathematics problems.
2. Review Calculus and apply it to realistic problems.
3. Use visualization tools to understand/present mathematics.
4. **Learn to do mathematical experiments.**
5. Master the software package for use in other courses/career.
6. Modelling (systems of ODEs) e.g. the SIR model.

Six experiments, one per assignment, requires programming.

1: The prime number race.

Prime Number Races by Andrew Granville and Greg Martin

Let $P_i = \{2, 3, 5, 7, ..., p_i\}$ where p_i is the i'th prime.
Define $s_i = |\{p \in P_i : p_i \equiv 1 \mod 4\}|$ and $t_i = |\{p \in P_i : p_i \equiv 3 \mod 4\}|$

p_i	2	3	5	7	11	13	17	19	23	29	31	37	41	547	7919	17389
s_i	0	0	1	1	1	2	3	3	3	4	4	5	6	47	495	986
t_i	0	1	1	2	3	3	3	4	5	5	6	6	6	53	504	1013

Does $s_i > t_i$ occur?
The first part of the question is to find the first prime for which $s_i > t_i$.

1: The prime number race.

```
> p := 2:
> s,t := 0,0:
> while s <= t do
>     p := nextprime(p);
>     if p mod 4 = 1 then s++; else t++; fi;
> od:
> p,s,t;
                                    26861, 1473, 1472
```

For primes less than 10^6 and determine for what %age of the cases is $s_i > t_i$, $s_i = t_i$ and $s_i < t_i$.
I get 0.194%, 0.0284%, and 99.7776%.

Another prime number race is to consider the primes mod 10. Which of 1,3,7,9 wins the race?

2: Checking theorems

Theorem $\int_{-\pi/2}^{\pi/2} \sin(mx)\cos(nx)dx = 0$ for all nonnegative integers $m > 0$ and $n > 0$.

(a) Use Maple to check it for $1 \leq m \leq 5$ and $1 \leq n \leq 5$.

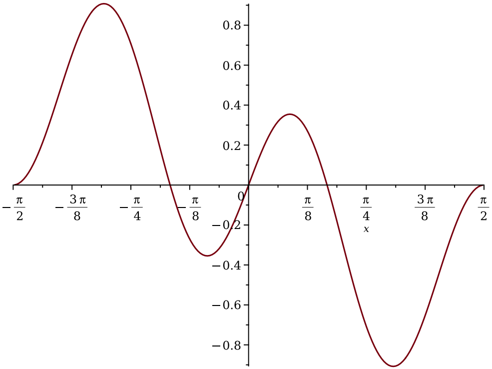
Here is some code

```
> for m from 1 to 5 do
>     for n from 1 to 5 do
>         print(m,n,int(sin(m*x)*cos(n*x),x=-Pi/2..Pi/2));
>     od;
> od;
                                              1, 1, 0
                                              1, 2, 0
                                              1, 3, 0
                                                ...
                                              5, 5, 0
```

I found this quite unsatisfactory. It provides no insight as to why $\sin mx$ and $\cos nx$ are orthogonal. So I now ask the students to graph some of the functions and explain why they are orthogonal.

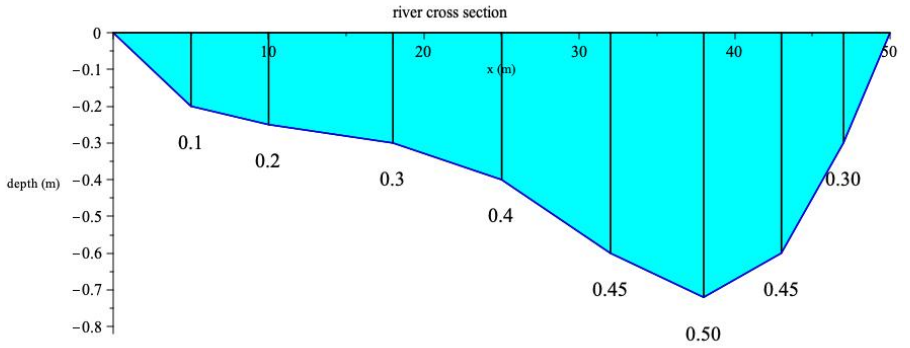
(b) Now graph $\sin(mx)\cos(nx)$ on $-\pi/2 \leq x \leq \pi/2$ for $m = 1, 2$ and $n = 1, 2$ and explain why they are orthogonal.

```
> plot(sin(3*x)*cos(2*x),x=-Pi/2..Pi/2 );
```

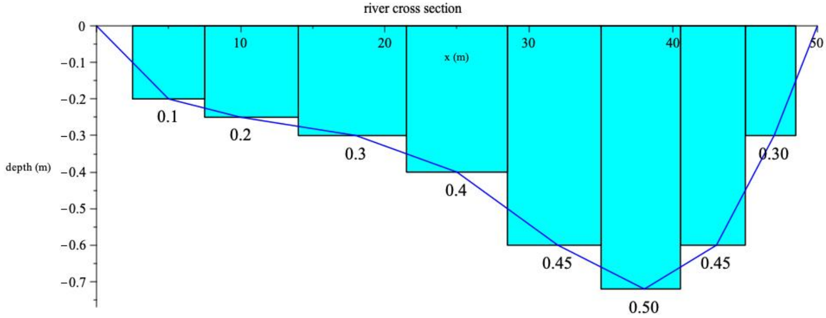



```
> combine(sin(3*x)*cos(2*x),trig);
                   1/2 sin(5 x) + 1/2 sin(x)
```

Measuring river flow (discharge)

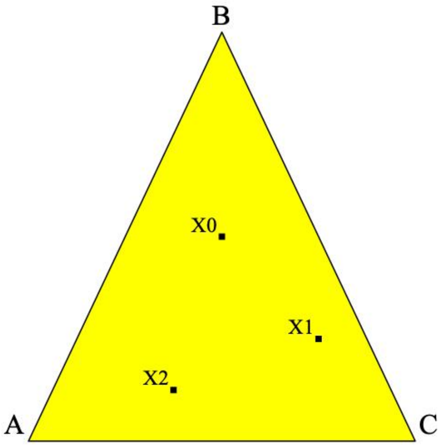


Depth $d(x)$ m for $a \leq x \leq b$.
Velocity $v(x)$ m/s for $a \leq x \leq b$.
Flow $\int_a^b v(x)d(x)dx\ m^3/s$.



Which approximation is best?
Need to know the exact value so use $d(x) = \sin(25/\pi x)$ and and $v(x) = 0.5\sin(25/\pi x)$.

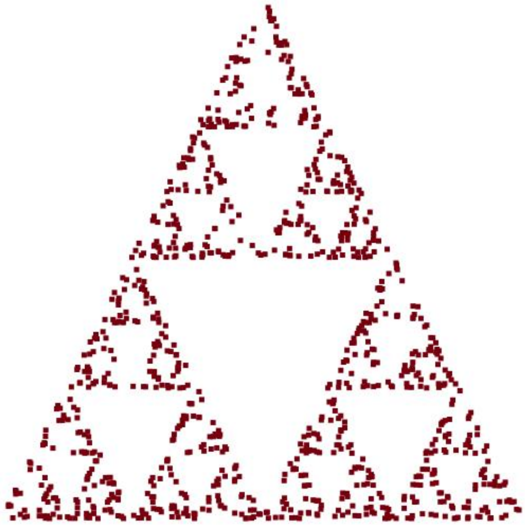
4: Random walks



Simulate the random walk, create a plot of the points and identify the image that you get.

```
> A,B,C := [0,0],[1,1],[2,0];
> n := 1000;
> Walk := Array(0..n):
> X := [0.5,0.5]:
> Walk[0] := X:
> R := rand(1..3): # R is a random number generator for {1,2,3}
> for i to n do
>   u := R();
>   if u=1 then X := (X+A)/2;
>   elif u=2 then X := (X+B)/2;
>   else X := (X+C)/2;
>   fi;
>   Walk[i] := X;
> od:
> L := convert(Walk,list):
> plot( L, style=point, symbol=solidbox );
```

4: Random walks



5: Visualizing eigenvalues and eigenvectors.

```
> A := Matrix([[1,1],[1,0]]);
```

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

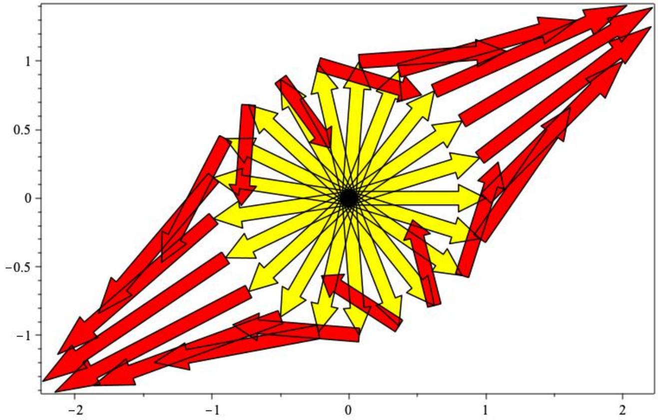
```
> with(LinearAlgebra):
> Eigenvectors(A);
```

$$\begin{bmatrix} \frac{\sqrt{5}}{2}+\frac{1}{2} \\ -\frac{\sqrt{5}}{2}+\frac{1}{2} \end{bmatrix}, \begin{bmatrix} \frac{1}{\frac{\sqrt{5}}{2}-\frac{1}{2}} & \frac{1}{-\frac{\sqrt{5}}{2}-\frac{1}{2}} \\ 1 & 1 \end{bmatrix}$$

```
> evalf(%);
```

$$\begin{bmatrix} 1.618033988 \\ -0.6180339880 \end{bmatrix}, \begin{bmatrix} 1.618033991 & -0.6180339890 \\ 1.0 & 1.0 \end{bmatrix}$$

5: Visualizing eigenvalues and eigenvectors.



$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

5: Visualizing eigenvalues and eigenvectors.

```
> with(plots):
> n := 21;
> A := Matrix([[1,1],[1,0]]);
> U := Array(1..n):
> V := Array(1..n):
> for i to n do
>    theta := 2.0*Pi*i/n;
>    u := <cos(theta),sin(theta)>;
>    v := A.u;
>    U[i] := arrow([0,0],u,width=0.1,color=yellow);
>    V[i] := arrow(u,v,width=0.1,color=red);
> od:
> L := convert(V,list),convert(U,list):
> display(L);
```

6: Simulating a 30 year mortgage payment

Suppose you take out a 30 year mortgage of \$200,000 at an annual interest rate of $r = 5\%$ compounded daily. Let $M(t)$ be the amount you owe on a 30 year mortgage at time t. That bank (should) determine the anual payment P\$/year such that $M(30) = 0$. We can approximate the process with a DE.

$$M'(t) = rM - P \quad \$/year$$

First solve the DE with the initial value $M(0) = 200000$ then solve for P such that $M(30) = 0$.

```
> r := 0.05; # annual interest rate
> de :=  diff(M(t),t) = r*M(t)-P;
> sol := dsolve( {de,M(0)=200000} );
```

$$sol := M(t) = 20P + (200000 - 20P)\,\mathrm{e}^{t/20}$$

```
> P := fsolve( eval(rhs(sol),t=30), P );
                              P := 12872.16917
> interest := 30*P-200000;
                                186165.0751
```

Now, starting with mortgage $M(0) = \$200,000$ simulate daily interest charges at a rate of $5\%/365$ per day (compounded daily) and monthly mortgage payments at a rate of $\$P/12$ per month for 30 years. Why do you not get to \$0?

```
> M := 200000;
> days := [31,28,31,30,31,30,31,31,30,31,30,31];
> for y to 30 do
>   for m to 12 do
>     for d from 1 to days[m] do M := M+r/365*M; od;
>     M := M-P/12; # pay on the last day of the month
>   od;
> od;
> M; # should be close to 0
                                 1733.447541
```

Repeat this with daily payments of $\$P/365$ then hourly interest charges and hourly payments. How close to $M(30) = 0$ to you get?

What lesson from Computer Science would be most useful?

$O(n^2)$ codes verses $O(n)$ codes

```
> GetListSquares := proc(n::nonnegint) local L,k;
>    L := [];
>    for k to n do L := [op(L),k^2]; od;
>    L;
> end:
>
> GetArraySquares := proc(n::nonnegint) local A,k;
>    A := Array(1..n);
>    for k to n do A[k] := k^2; od;
>    convert(A,list);
> end:
>
> st := time(): GetListSquares(10^5): time()-st;
                                       14.566
> st := time(): GetArraySquares(10^5): time()-st;
                                        0.078
```

The prime number race

Let $P_i = \{2, 3, 5, 7, ..., p_i\}$ where p_i is the i'th prime.
Define $S_i = \{p \in P_i : p_i \equiv 1 \mod 4\}$ and $T_i = \{p \in P_i : p_i \equiv 3 \mod 4\}$
Does it ever happen that $|S_i| > |T_i|$?

Many students constructed the sets S_i and T_i like this.

```
> p := 2:
> S,T := {},{}:
> while nops(S) <= nops(T) do
>     p := nextprime(p);
>     if p mod 4 = 1 then S := S union {p}; else T := T union {p}; fi;
> od:
> i,p,nops(S),nops(T);
                                  2945, 26861, 1473, 1472
```