Mathematica Project, Math21b Spring 2008

Oliver Knill, Harvard University, Spring 2008

Support

Welcome to the Mathematica computer project! In case problems with this assignment, please email Oliver at math21b@fas.harvard.edu. *Mathematica* can demand a lot of resources from computers. Therefore, **save the notebook frequently** during your work. The actual assignments are in the last blue box of this notebook. This lab is due at the last lecture of the semester. You will have to print out the assignment part with your solutions and hand them in. All problems are listed at the end. Turn in 3 of the 5 problems to get full credit. All problems have sample code in this notebook somewhere. The total work should be minimal. It can be done in a 2-3 hours if you are in a hurry. If you think you need too much time, network around or ask. This project is not intended to take a large amount of time. Check the syllabus to remind yourself how the lab is used for your grade.

Cells and Expressions

The notebook is subdivided into cells, bounded by a blue bracket. This is a "text" cell, the next is an "input" cell. We compute the dot product of two vectors:

```
{3, 4, 5, 6}.{2, 3, 4, 5}
```

To evalute an expression, click anywhere on the expresson, hold down the <Shift> key and press <Enter>. This is how we compute the inverse of a matrix:

```
Inverse[{{3, 2, 5}, {3, 2, 1}, {1, 1, 1}}]
```

In this notebook, input cells are boxed in green for better readability.

Basic Computations

Mathematica allows to work with vectors and matrices in the same way as with numbers. A dot . is the matrix multiplication or dot product. The following examples should be selfexplanatory.

```
{{1, 4}, {0, 1}}.{2, 3}
```

```
A = {{2, 3}, {1, 2}}; b = {3, 4}; Inverse[A].b
```

```
Solve[{x + y == 1, x - y == 3}, {x, y}]
```

```
Det[{{2, 3, 5, 6}, {1, 2, 3, 4}, {1, 1, 1, 1}, {12, 3, 2, 1}}]
```

```
Tr[{{a, b}, {c, d}}]
```

```
MatrixForm[Transpose[{{1, 2, 3}, {3, 4, 5}, {1, 2, 1}}]]
```

```
A = {{1, 2}, {3, 4}}; QR = QRDecomposition[A];
Q = QR[[1]]; R = QR[[2]]; {MatrixForm[Q], MatrixForm[R]}
```

Eigenvalues

With Eigenvalues[A], we can compute the eigenvalues, with Eigensystem[A], both the eigenvalues and the eigenvectors. What happens for a random matrix? Here, we plot the complex eigenvalues of a random 50x50 matrix. You might want to try with larger matrices to see a pattern.

```
A = Table[Random[] - 1 / 2, {50}, {50}];
eigenvalues = Eigenvalues[A];
point[z_] := Point[{Re[z], Im[z]}];
Graphics[Map[point, eigenvalues]]
```

Explore this for larger and larger matrices. What do you observe?

An other question. Sergey Sadov of the Memorial university in Canada asked: Suppose A is an NxN matrix with entries independently and uniformly distributed on[0, 1].What is the distribution of the absolute value of the largest eigenvalues of A for large N?

```
le := Module[{},
   A = Table[Random[], {100}, {100}]; Max[Abs[Eigenvalues[A]]]];
```

Lets do an experiment, take 1000 matrices, compute the largest eigenvalue in each case and look at the distribution. This computation takes some time.

```
s = Sort[Table[le, {1000}]];
```

If we plot these data, we see the distribution function:

```
ListPlot[s]
```

The density function can be visualized as follows:

```
ListPlot[BinCounts[s, {Min[s], Max[s], 0.1}], Ticks → None]
```

While we are at it, lets also compute some statistical properties:

```
{Mean[s], Median[s], StandardDeviation[s], Min[s], Max[s]}
```

Determinants

What is the distribution of the determinant of a random matrix? To see the distribution of a random 10x10 matrix, lets do 100 experiments and plot the results in a sorted manner.

```
S = Sort[Table[Det[RandomReal[{0, 1}, {10, 10}]], {100}]];
ListPlot[S]
```

This distribution approaces the arcsin distribution.

```
Plot[ArcSin[x], {x, -1, 1}]
```

Here is how we compute the determinant of the 100 x 100 matrix which contains the entry i*j at the entry (i,j).

```
A = Table[i * j, {i, 100}, {j, 100}];
Det[A]
```

Differential Equations

Differential equations can be solved with the command "DSolve". Here is an example, where one does not specify the initial conditions:

```
DSolve[f'''[x] - 2 f''[x] + f'[x] == Exp[3 x], f[x], x]
```

Here is an example, where the initial conditions are specified. This is the solution of the forced driven harmonic oscillator:

```
S = DSolve[{f''[x] + f'[x] + f[x] == Cos[x],
   f[0] == 10, f'[0] == 0}, f[x], x]
```

```
g[t_] := S[[1, 1, 2]] /. x → t; Plot[g[t], {t, 0, 20}]
```

Markov Processes

A vector is called a probability vector if each entry is nonnegative and the sum of all its entry is 1. A matrix A for which each column is a probability vector is called a Markov matrix or stochastic matrix.

```
A = {{1/2, 1/5, 1/4, 1/3}, {1/6, 1/5, 1/4, 0},
   {1/6, 1/5, 1/4, 1/3}, {1/6, 2/5, 1/4, 1/3}};
MatrixForm[
 A]
```

A Markov matrix has an eigenvalue 1 because its transpose has the eigenvector [1,1,..., 1]. The eigenvector of the eigenvalue 1 of A is called the steady state.

```
N[Eigensystem[A]]
```

When iterating the matrix A, it converges to a matrix which has the eigenvector as column vectors.

```
N[MatrixPower[A, 10]]
```

Here is a magical square:

```
A = {{17, 24, 1, 8, 15}, {23, 5, 7, 14, 16},
   {4, 6, 13, 20, 22}, {10, 12, 19, 21, 3}, {11, 18, 25, 2, 9}};
MatrixForm[
 A]
```

If we divide it by the sum 65 over each column, we obtain a stochastic matrix.

Products of random matrices

If we multiply random 2x2 matrices, how fast does the product grow? The answer depends on the size of the random numbers. Lets take random numbers between -c and c:

```
F[c_] :=
 Module[{}, randommatrix := Table[c (2 Random[] - 1), {2}, {2}];
  s = IdentityMatrix[2]; lambda = 0;
  Do[s = randommatrix.s; u = Max[Abs[Flatten[s]]];
   s = s / u; lambda = lambda + Log[u], {100}]; lambda]
```

```
Plot[F[c], {c, 0.5, 3}]
```

For which range of c, does the product appear to grow exponentially?

Nonlinear Systems

Here is an example, of how to plot a phase portrait of a nonlinear system. It is a Volterra system which is a model in population dynamics.

```
Needs["VectorFieldPlots`"];

p[x_, y_] := 0.4 * x - 0.4 * y * x;
q[x_, y_] := -0.1 * y + 0.2 * x * y;

Module[{s = {}, a = 0.45, b = 1},
 Do[S = NDSolve[{x'[t] == p[x[t], y[t]], y'[t] == q[x[t], y[t]],
     x[0] == a, y[0] == b + j / 10}, {x, y}, {t, 0, 40}];
  g[x_, y_] := {p[x, y], q[x, y]};
  s = Append[s,
    ParametricPlot[Evaluate[{x[t], y[t]}] /. S[[1]], {t, 0, 40},
     PlotRange → {{0, 2}, {0, 2}}, Axes → True, Frame → False,
     PlotStyle → {RGBColor[0.3, 0.3, 1]}, DisplayFunction → Identity,
     Frame → False, Axes → False, Ticks → None]], {j, 12}];
 s2 = VectorFieldPlot[g[x, y], {x, 0, 2}, {y, 0, 2},
   ColorFunction → Hue, ColorOutput → RGBColor,
   DisplayFunction → Identity]; Show[{s, s2}]]
```

Fourier Series

Lets compute some Fourier series. In this example, we compute the series of an odd function so that we only have to compute the sin-series.

```
ClearAll[f, a, aa, s]
f[x_] := If[x < 0, 1, -1];
a[n_] := Integrate[(1 / Pi) f[x] Sin[n x], {x, -Pi, Pi}];
aa = Table[a[k], {k, 1, 30}];
s[x_, n_] := Sum[aa[[k]] Sin[k x], {k, 1, n}];
```

For every n, we have a Function s[x,n] which is a Fourier approximation of the function f:

```
Plot[{s[x, 5], f[x]}, {x, -Pi, Pi}]
```

If you plot several plots, you observe that while the Fourier approximation converges pointwise, there is an overshoot near the discontinuity. This is called the Gibbs phenomenon.

```
Plot[{f[x], s[x, 1], s[x, 20], s[x, 30]}, {x, -Pi, Pi}]
```

Partial Differential Equations

Here is how *Mathematica* can solve a partial differential equation. Here it is the wave equation. The result will be given in terms of variables C[1],C[2] which can be general functions:

```
DSolve[{D[u[x, t], {t, 2}] == D[u[x, t], {x, 2}]}, u, {x, t}]
```

Note that the solution capabilities of *Mathematica* are rather limited in partial differential equations.
Here is a numerical computation of the one dimensional wave equation with inhomogeneous right hand side on the unit interval with boundary conditions zero at 0 and 1.

```
s = NDSolve[{∂{t,2} u[x, t] == ∂{x,2} u[x, t] - 20 Sin[x],
    u[x, 0] == Sin[6 π x], u^(0,1)[x, 0] == -6 π Cos[6 π x],
    u[0, t] == 0, u[1, t] == 0}, u, {x, 0, 1}, {t, 0, 1}];
Plot3D[Evaluate[u[x, t] /. s[[1]]], {t, 0, 1}, {x, 0, 1}]
```

Sound and Fourier

Instead plotting a function, you can also play it. A song is just a function.

```
Play[ Sin[1000 Sqrt[x]], {x, 0, 2}]
```

If you superimpose several functions, you can hear several "melodies" together.

```
w = Play[
  Abs[Sin[200 Sin[x^2]]] + Sin[1000 x] + Cos[2000 Sqrt[x]], {x, 0, 6}]
```

You can export it as a Wave file and use it as a cellphone ring tone ....

```
Export["sound.wav", w, "WAV"]
```

The smoothness of a function depends on how nice it sounds. A discontinuous function sounds horrible. The fourier series decays slowely.

```
Play[ Sign[Sin[1000 * x]], {x, 0, 2}]
```

Sound generation

Here is how you can play a tune with *Mathematica* with a given wave form. In the example below, we take the Sin function as the wave form and play a tune.

```
PlaySong[hull_, tune_, ground_] :=
 Module[{soundfilename, scale, songlength, frequency, song, P},
  scale[n_] := ground * 2^(n / 12); beatlength = 1 / 5;
  songlength = Length[tune] * beatlength;
  frequency[x_] := tune[[1 + Floor[x / beatlength]]];
  song[t_] := hull[scale[frequency[t]] * t];
  P = Play[song[t], {t, 0, songlength}]]

t1 = {0, 0, 0, 2, 1, 2, 1, 2, 1, 2, 0, 0, 0, 4,
   10, 7, 7, 5, 4, 2, 2, 0, 4, 7, 4, 0, 0, 0, 0, 0, 2, 0, 0};
f1[x_] := Abs[Sin[x]]; g1 = 2000;
PlaySong[f1, t1, g1]
```

Random music

Lets play some random tune:

```
T[x_] := Module[{}, y = x + RandomInteger[4] - 2; y];
s = NestList[T, 5, 100];
song1 = Sound[SoundNote[#, 1, RandomChoice[{"Cello"}]] & /@ s, 20]
```

You can export this into a midi file:

```
Export["song1.midi", song1, "Midi"]
```

here is an example with random accords. We also change time randomly.

```
r := RandomChoice[{4, 7, 12}]
g[x_] :=
  RandomChoice[{0.5, 0.3, 0.2} → {x, {x[[1]]}, {x[[1]], x[[2]]}}];
accord := Module[{}, r1 = r; {{1 + r1, 5 + r1, 9 + r1}, {1, 5, 9}}];
scale = {1 + r, 5 + r, 9 + r};
s := g[RandomChoice[accord]] + RandomChoice[scale]
A = Table[{s, Random[Integer, 3] + 1}, {40}];
sn[{a_, t_}] := SoundNote[s, t];
song2 = Sound[{"Piano", sn /@ A}, 12]
```

```
Export["song2.midi", song2, "Midi"]
```

The *Mathematica* documentation gives the following cool example of a sound generation using cellular automaton 30:

```
sound3 =
 Sound[SoundNote[DeleteCases[3 Range[31] Reverse[#], 0] - 48, .1] & /@
   Transpose[CellularAutomaton[90, {{1}, 0}, 30]]]
Export["sound3.midi", sound3, "MIDI"]
```

Assignment

To get full credit for this assignment, you have to hand in solutions to the following problems. We kept the assignment short. Once you get again a grip at *Mathematica*, it should take you not long to do it. You have sample code for each of the problems in the text above. You need to turn in answers to three of the following 5 problems to get full credit:

1) What is the distribution of the eigenvalues of a random matrix, where each entry is a random number between -1,1. Make experiments with larger matrices (sample code is above) and write down your observation in a sentence. How does the eigen-value cloud look like in your experiments?

2) Find the solution of the ordinary differential equation f''[x]+f'[x]+f[x]== Cos[x]+x^2with initial conditions f[0]==1;f'[0]== 0.

3) Find the Fourier series of the function f[x_]: = x^5 on [-Pi,Pi]. You need at least 20 coefficients of the Fourier series.

4) When multplying random 2x2 matrices where the entries are random numbers in [-c,c]. There is a threshold of c, for which the product starts to grow exponentially. Where is this treshold?

5) Write a little music tune using the above music routines. If you have produced a MIDI file you want to share, send it to Oliver. (No worry, you keep the rights of the song... and all loyalties if it should become a hit.)

If you find something cool during your experiments, feel free to include it also. In order to work on your project it is a good idea to save this notebook first as a different document, do the assignment directly in that notebook and print out the relevant pages at the very end on a printer. The assignments have to be printed out and turned in the last class. The printout does not have to be in color but feel free to print it in color, if you like.