Handout 6   Modern Coding Theory
Homework 8   May 14, 2008

PROBLEM 1. The purpose of this homework is to use the Wormald method to study a model for "preferential attachment."

Consider n nodes. Initially all nodes have degree 0. Assume that we allow a maximum degree of d_{max}. We proceed as follows. At every step pick two nodes independently from the set of all nodes which have degree at most $d_{max}-1$. Rather than picking them with uniform probability pick them proportional to their current degree. More precisely, assume that at time t you have $D_i(t)$ nodes of degree i. Then pick a node of degree i with probability

$$\begin{cases} \frac{D_i(i+1)}{\sum_{j=0}^{d_{max}-1} D_j(j+1)}, & 0 \leq i < d_{max}, \\ 0, & i = d_{max}. \end{cases}$$

Clearly, at time $nd_{max}/2$, all nodes will have maximum degree.

Pick $d_{max} = 4$.

- Write down the set of differential equations for this problem. Are the conditions fulfilled?

- Plot the evolution of the degree distribution as a function of the normalized time for $\tau \in [0, 1]$, $\tau = t/n$.

- Simulate this problem and plot on the same plot some simulation runs for $n = 10^3$ as well as $n = 10^4$.

HINT: In general one can not expect to solve the system of differential equations analytically. But it is typically easy to solve them numerically. Here is how you do it in Mathematica. The following lines set up the differential equation we discussed in class and plots the solution.

```
cnds = {n[0] == 1/3};                          (* initial conditions *)
eqns = {n'[u] == -2 n[u]^2};                   (* set of diff equations *)
eqnspluscnds = Flatten[Join[eqns, cnds]];      (* put the two together *)
umax=10;                                        (* solve equation up to this po
sol = Flatten[NDSolve[eqnspluscnds, {n}, {u, 0, umax} ]] (* solve the diff equation *)
Plot[Evaluate[{n[u]} /. sol], {u, 0.0, umax}]   (* plot the solution *)
```

If you have more than one variable then it is often convenient to call them

```
d[0][u], d[1][u], d[2][u], ...
```

In this case you might have something like

```
cnds = {d[0][0] == ..., d[1][0]==..., ...};
eqns = {d[0]'[u] == ..., d[1]'[u]==..., ...};
eqnspluscnds = Flatten[Join[eqns, cnds]];
umax=...;
sol = Flatten[NDSolve[eqnspluscnds, {d[0], d[1], ...}, {u, 0, umax} ]]
Plot[Evaluate[{d[0][u], d[1][u], ...} /. sol], {u, 0.0, umax}]
```