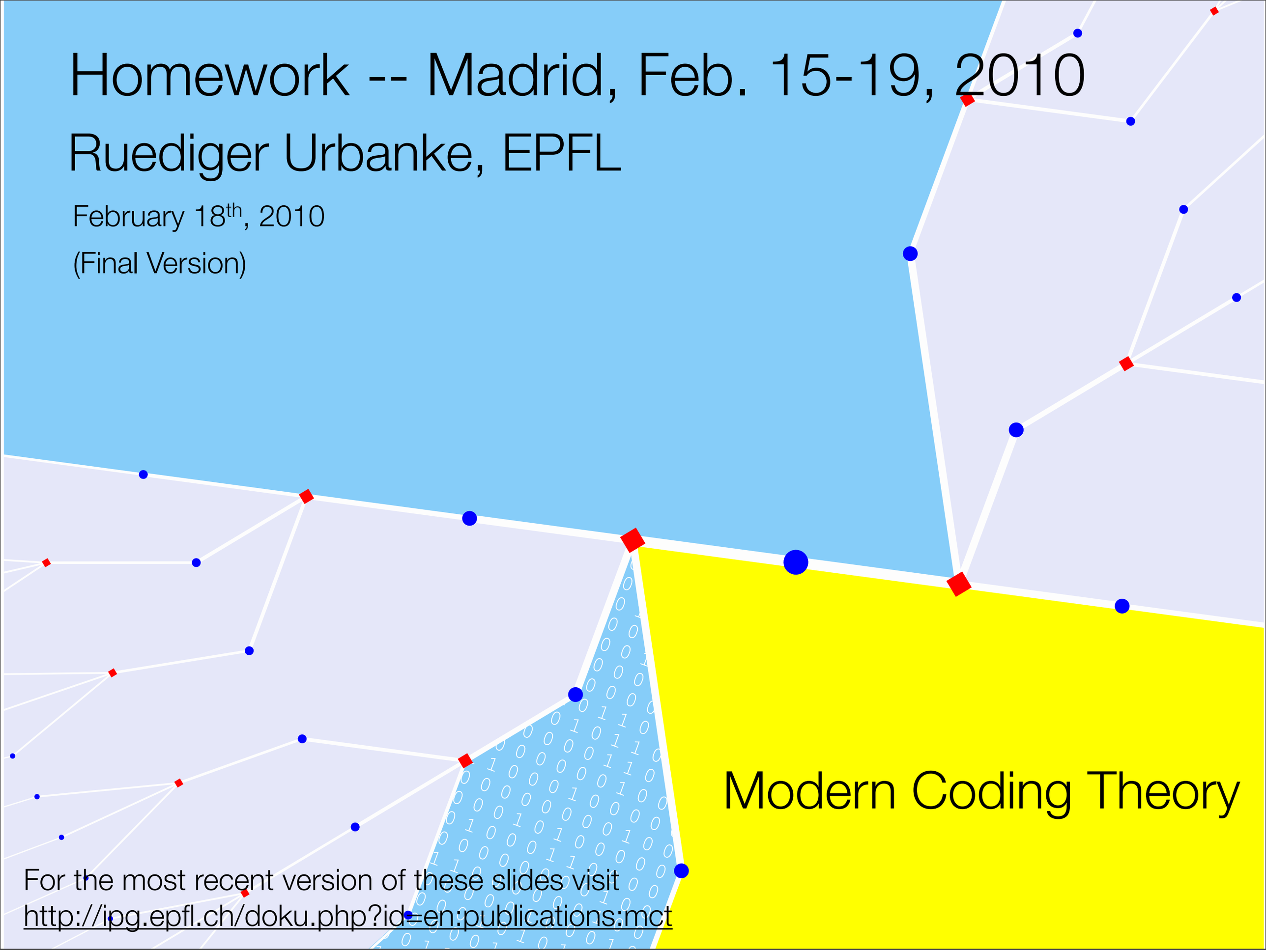
Homework -- Madrid, Feb. 15-19, 2010

Ruediger Urbanke, EPFL

February 18th, 2010

(Final Version)



Instructions

The following pages contain homework problems for the course *Modern Coding Theory*, Madrid, February 15-19, 2010.

Rules:

1. You can discuss these problems with your friends (assuming you have any).

2. When you write down the solution, write it down yourself. No copying, cutting, or pasting.

3. I do not expect that you solve all problems. Also, some problems are harder than others, and some questions are open ended. Pick some that interest you and solve them. Write to the point. I will not weigh your answers, I will correct them!

Enjoy!

I. Introduction

I-1. There are many matrices G that all define the same code.
For an (n, k) binary code, how many such “equivalent” matrices are there?

I-2. Show that the MAP decoder minimizes the probability of error.
This is why we would like to implement it.

II. Factor Graphs

II-1. A large group of people take a pleasant walk in the countryside. Unfortunately, it suddenly gets very foggy. Perhaps someone got lost? If we could just count all the people in the group ...

Every person can only see the people “closest” to him/her. Seen from above, the whole group does not walk in a single line, but the corresponding “graph” of “neighbors” forms a tree. Devise a simple message-passing algorithm so that after a finite number of messages have been passed all members of the group know the total size.

II-2. Consider the simple binary (7, 4) Hamming code. Draw the corresponding factor graph. Pick a non-zero codeword from C, call it x. Sent x over a binary erasure channel. Pick two positions and erase them. Show that the BP algorithm simplifies to rules discussed in class when transmission takes place over the BEC. Using this algorithm, try to recover the codeword.

Extra credit: Does this algorithm always succeed for the Hamming code assuming that there are at most 2 erasures? Prove or give a counter-example. For the present setting, how many erasures can an optimal algorithm recover from?

III. Ensembles

III-1. Generate random samples of a (3, 6) ensemble of length n according to the configuration model which we discussed in class. Write a small program (C or Matlab) to accomplish this.

III-2. For n=2^m, m=6, 8, 10, generate random samples and compute the rank. In particular, estimate the probability that a random sample has full rank.

III-3. Produce an n x n dense matrix by picking each component equally likely to be 0 or 1. Numerically check its rank. For increasing n, estimate numerically the probability that the matrix is full rank.

Extra credit: What is the chance that this matrix is full rank?

VI. Analysis

We want to construct a “good” code of rate 3/4, both in the asymptotic sense but also for moderate lengths.

VI-1. Use any method you want find a degree distribution of rate 3/4 and with a large threshold when transmission takes place over the BEC. [No need to go crazy when doing the optimization. You might just want to guess.] Describe how you found the degree distribution.

VI-2. On page 144 in the book (see web page for copy) you find in Conjecture 3.152 an analytic expression for the finite-length behavior of such a code. This expression only depends on parameters that you can compute with density evolution. Use this expression and plot the block error probability for n=1024 for the degree distribution you found in part 1. If you had to come up with a very good degree distribution of rate 3/4 and n=1024 (and had unbounded enthusiasm and time), what would you do.