The third project is individually assigned. Topics are listed below. For each topic we list at least one paper and formulate questions. This sheet confirms and clarifies earlier assignments.

1) δ-GCD versus ϵ-GCD

In the lectures the GCD of polynomials with approximate coefficients we considered is known as the ϵ-GCD. Matching the common roots, there is the notion of δ-GCD, defined in

- V. Pan: **Computation of approximate polynomial GCDs and an extension.** *Information and Computation* 167:71–85, 2001.

The goal of this project is to compare the δ-GCD with the ϵ-GCD. For which instances to they agree with each other, and when do they disagree? What are their advantages and disadvantages?

2) the multiplicity structure of an isolated root

Given the exact (or sufficiently accurate) location of an isolated multiple root, can one *locally* determine the multiplicity of the root? The book contains a sketch of an algorithm, also presented in

- Hans J. Stetter: **Analysis of zero clusters in multivariate polynomial systems.** In *Proceedings of 1996 International Symposium on Symbolic and Algebraic Computation*, edited by Y.N. Lakshman, pages 127-136, ACM 1996.

To test an implementation in Maple, you may assume the multiple root is located at the origin and defined by monomial ideals, so that you can easily generate zeroes for which the multiplicity is known a priori.

3) comparing two implementations of mixed volumes

As we went deeper in the book, it became clear how central the "BKK bound" became. Two recent very efficient implementations of mixed are publicly available:

- Tangan Gao and T.Y. Li: **Mixed volume computation for semi-mixed systems.** *Discrete Comput. Geom.*, Volume 29, Number 2, pages 257-277, 2003. Software available at http://www.csulb.edu/~tgao and http://www.math.uic.edu/~li.
- Gunji, T. and Kim, S. and Kojima, M. and Takeda, A. and Fujisawa, K. and Mizutani, T.: **PHoM – a polyhedral homotopy continuation method for polynomial systems.** *Computing*, Volume 73, pages 55-77, 2004. Software available at http://www.is.titech.ac.jp/~kojima/PHoM/index.html.

The obvious question is: which software is most efficient? Make sure you try at least a dozen examples to come to a conclusion. A harder question is to find reasons why one software outperforms the other on certain examples.

4) multivariate approximate GCD

In lecture 18 we outlined the recent method of Zhonggang Zeng to compute approximate GCD's of univariate polynomials. There is a sequel to this paper:

- Zhonggang Zeng and Barry H. Dayton: **The approximate GCD of inexact polynomials part II: A multivariate algorithm.** In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 320-327, ACM 2004.

Your technical report should summarize the paper: define the problem, the method, do a computation, and write a conclusion. Is this problem much harder than computing the GCD of univariate polynomials?

5) application of GCD to image processing

As advertised in the beginning of the course, the approximate GCD is very relevant to practical scientific computing, as illustrated by the following paper:

- S. Unnikrishna Pillai and Ben Liang: **Blind Image Deconvolution Using a Robust GCD Approach.** *IEEE Transactions on Image Processing*, Volume 8, Number 2, pages 295-301, 1999.

An important aspect to consider in this project is the definition of the problem. How does one get to needing an approximate GCD? Define the original problem and illustrate.

6) automatic differentiation

The cost of a numerical root finder is determined largely by the cost of evaluation and differentiation. The following paper describes a method to evaluate sparse Jacobian matrices efficiently:

- Thomas F. Coleman and Arun Verma: **The efficient computation of sparse Jacobian matrices using automatic differentiation.** *SIAM J. Sci. Comput.*, Volume 19, Number 4, pages 1210-1233, 1998.

Describe the method via some well chosen examples. Do the authors claim their method is optimal?

recommendations for the report

Go to google, type in "LaTeX tutorial", and click "I am feeling lucky". That should bring you to the URL http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/ of the online tutorial of David R. Wilkins.

Please keep the size of the appendix limited. Instead of a very long listing of Maple experiments, it is better to list the Maple code of the experiment which would allow the reader to rerun the calculations which produced the numbers in the report.

Deadlines are Wednesday 24 November 2004 at 1PM or Wednesday 8 December 2004 at 1PM

In principle, for those taking the final exam, the project must be handed in on Wednesday 24 November 2004. If the final exam is to be replaced by a project, then you should invest more time in the project and then you have time till Wednesday 8 December. The first deadline should be considered as a target date – you can miss it, but you should strive to hit it – the second deadline is a hard deadline.

As with project two, you should hand in a technical report between five and ten pages long. There are three items the report must contain.

In this report you must define the problem and describe *in your own words* the results of the paper(s). When copying from the paper(s) – e.g., the statement of a theorem – you must explicitly cite the paper.

Secondly, the report should describe a computational experiment illustrating the most important aspects discussed in the paper. An appendix to the report can be the output of a Maple worksheet, or the listing of a program on some examples.

The third and last element of the report is the conclusion. This should read like an executive summary. It should be brief, but completely self-contained. Summarize in one or two paragraphs what is most important about your project.

Feel free to come to my office for help.