Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Prof. R. Hiptmair 4. Februar 2010

Numerical Methods for CSE – HS 2009

Examination

Surname : ....
First name : ....
Legi-number : ....
Computer-number : slabhg ....
Signature : ....

Important instructions

- Keep calm!
- Cell phones and other communication devices are not allowed. Make sure that they are turned off.
- First, fill in the above lines and put your legimitation card onto the desk.
- Please login: login-name: **exam**, password: **ethz**
  Into the additional box fill in your n.ethz-login.
- Open a terminal (by right click and choosing console). Start up acroread (`acroread &`) and MATLAB (`matlab &`) and open the lecture notes located in the directory `resources/Documents/`. There are two versions (full-sided and 4-by-one).
- The exam starts when all students have successfully logged in and started up Matlab. The exam is then distributed. As examination language you can choose either english or german. The exam lasts 180 minutes.
- Get a general idea of the exercises. Pay attention to the number of points awarded for each subtask. It is roughly correlated with the amount of information your answer should contain.

Problem	Max. points	Reached points	Visa
1	**6**		
2	**12**		
3	**22**		
4	**18**		
5	**12**		
Total	**70**		

- Begin each exercise with a new sheet of paper and label it with your name and the number of the exercise on the top right.
- *Write clearly and use no pencil*. Per exercise at most one attempt can be handed in. Cross invalid attempts clearly out.

Matlab concerning instructions

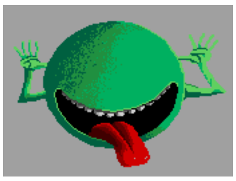
- The programing templates as well as the script are located in the directory `resources`. You are not allowed to use other resources.
- **Save all digital results (codes, plots) in the directory `'results'`!** Only the results in this directory are being corrected!
- Save plots as **.eps** files.
- Label each Matlab file with your name (as comment, at the top of the file).
- In MATLAB Windows-like shortcuts can be used (apply it in Files→Preferences→Keyboard→Active settings)

At the end of the examination

- Your written results are being collected together with the instruction and problem sheets. During the exam the digital results are stored every quarter to a save location.
- Do not log out and don't turn off the computer!

Problems

- If you have problems with your operation system please rise your hand to get support.

 Good luck!

Prof. R. Hiptmair

Examination

February 4th, 2010

Problem 1: Efficient matrix multiplication (6 points)

Given the fully populated matrices $\mathbf{A} \in \mathbb{R}^{n,2}$, $\mathbf{B} \in \mathbb{R}^{2,n}$, $\mathbf{C} \in \mathbb{R}^{n,2}$, $n \gg 2$, determine the asymptotic complexity (in terms of the problem size parameter n) of the evaluation of the MATLAB expressions `(A*B)*C` and `A*(B*C)`. Explain your answer.

Problem 2: Direct power method (12 points)

The following is known about the large sparse matrix $\mathbf{A} \in \mathbb{R}^{n,n}$, $n \in \mathbb{N}, n \gg 5$:

- $(\mathbf{A})_{i,i} = 5$ for all $1 \leq i \leq n$,
- $|(\mathbf{A})_{i,j}| \leq 1$ for all $1 \leq i < j \leq n$,
- each row of $\mathbf{A}$ has at most four non-zero entries,
- $\mathbf{A}$ is symmetric.

a) **(2 points)** Show that the matrix $\mathbf{A}$ is positive definite.

b) **(4 points)** Implement an efficient MATLAB function

```
Hv = H_times_v(A,u,v).
```

that computes $\mathbf{Hv}$ for the matrix $\mathbf{H} := \mathbf{A} + \mathbf{u}\mathbf{u}^T$, $\mathbf{u} \in \mathbb{R}^n$, and a vector $\mathbf{v} \in \mathbb{R}^n$. What is the asymptotic complexity of your routine in terms of the problem size parameter n?

c) **(6 points)** Implement a MATLAB function

```
lmax = dir_pow_meth(A,u,tol)
```

that computes an approximation of the largest eigenvalue of $\mathbf{H}$ by means of the direct power method. The function should make use of `H_times_v(A,u,v)` from subtask b). Use $\mathbf{u}$ as initial guess for the corresponding eigenvector. The iteration should be stopped once the relative change of the eigenvalue approximation drops below the tolerance `tol`.

Problem 3: Saddle point problem (22 points)

We consider a large block-partitioned linear system of equations

$$\mathbf{M}\mathbf{x} = \mathbf{b} \in \mathbb{R}^n \quad \text{with} \quad \mathbf{M} := \begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \;, \quad \mathbf{b} := \begin{pmatrix} \mathbf{c} \\ 0 \end{pmatrix} \;, \tag{1}$$

where

$$\mathbf{A} \in \mathbb{R}^{n,n} \quad \text{s.p.d. and tridiagonal} \quad , \quad \mathbf{B} \in \mathbb{R}^{m,n} \;, \quad \mathbf{c} \in \mathbb{R}^n \;,$$

and $n > m \gg 1$.

We know that each row of $\mathbf{B}$ has no more than two non-zero entries.

a) **(2 points)** Show that $\mathbf{M}$ is not positive definite.

b) **(3 points)** Give a sharp bound for the cardinality $\sharp\,\mathrm{env}(\mathbf{M})$ (i.e., number of tupels contained in it) of the envelope $\mathrm{env}(\mathbf{M})$ of $\mathbf{M}$.

c) **(3 points)** Use the MATLAB `spy` command to visualize the non-zero entries of $\mathbf{A}^{-1}$ for the tridiagonal matrix

$$\mathbf{A} = \begin{pmatrix} 3 & 1 & 0 & & \dots & & 0 \\ 1 & 3 & 1 & & & & \vdots \\ 0 & 1 & 3 & \ddots & & & \\ \vdots & & \ddots & \ddots & & & \vdots \\ & & & & \ddots & \ddots & 0 \\ \vdots & & & & \ddots & 3 & 1 \\ 0 & & \dots & & 0 & 1 & 3 \end{pmatrix} \in \mathbb{R}^{10,10} \;.$$

To that end write a short MATLAB script `vis_A_pattern.m`.

Hint: The complete inverse of a regular matrix is provided by the MATLAB function `inv(A)`.

d) **(3 points)** We introduce a partitioning of the solution vector of (1) according to

$$\mathbf{x} = \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \quad , \quad \mathbf{y} \in \mathbb{R}^n \;, \quad \mathbf{z} \in \mathbb{R}^m \;.$$

Assume that $\mathbf{M}$ is regular. Derive a Schur complement system $\mathbf{S}\mathbf{z} = \mathbf{q}$, $\mathbf{S} \in \mathbb{R}^{m,m}$, $\mathbf{q} \in \mathbb{R}^m$, whose solution supplies the $\mathbf{z}$-component of the solution $\mathbf{x}$ of (1).

Hint: Just eliminate the $\mathbf{y}$-component of $\mathbf{x}$ in the partitioned linear system

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ 0 \end{pmatrix} \;. \tag{2}$$

Alternatively, you may rely on block Gaussian elimination.

e) **(4 points)** The symmetric tridiagonal matrix $\mathbf{A} \in \mathbb{R}^{n,n}$ can be specified through its diagonal $\mathbf{d} \in \mathbb{R}^n$ and first sub- and super-diagonal $\mathbf{r} \in \mathbb{R}^{n-1}$:

$$\mathbf{A} = \begin{pmatrix} d_1 & r_1 & 0 & & \dots & & 0 \\ r_1 & d_2 & r_2 & & & & \vdots \\ 0 & r_2 & d_3 & \ddots & & & \\ \vdots & & \ddots & \ddots & & & \vdots \\ & & & & \ddots & \ddots & 0 \\ \vdots & & & & \ddots & d_{n-1} & r_{n-1} \\ 0 & & \dots & & 0 & r_{n-1} & d_n \end{pmatrix}$$

Devise an efficient MATLAB routine

```
S = Schurcomplement(d,r,B)
```

that computes the Schur complement matrix $\mathbf{S} \in \mathbb{R}^{m,m}$ found in sub-problem d). Here, `d,r` pass the column vectors $\mathbf{d}, \mathbf{r}$, and `B` the sparse $m \times n$-matrix $\mathbf{B}$.

Hint: Remember the initialization routine `spdiags` for initializing sparse banded matrices. You may use the MATLAB \-operator.

f) **(1 point)** What is the asymptotic complexity of your implementation of `Schurcomplement` w.r.t. problem size parameters n, m?

Hint: You may assume that MATLAB makes optimal use of the fact that $\mathbf{A}$ is tridiagonal and s.p.d.

g) **(6 points)** Implement an efficient MATLAB function

```
z = solveSchurcomplement(d,r,B,c)
```

that solves the Schur complement system $\mathbf{S}\mathbf{z} = \mathbf{q}$ from sub-problem d) iteratively by means of the (non-preconditioned) conjugate gradient method provided through MATLAB's `pcg` built-in function. Use the default tolerance of `pcg` and initial guess 0.

The function arguments `d,r,B` have the same meaning as for `Schurcomplement` from sub-problem e), and `c` contains the column vector $\mathbf{c} \in \mathbb{R}^n$, see (1).

Problem 4: Non-linear least squares (18 points)

The tuples (t_i, a_i), $1 \leq i \leq n$, represent measurements of the activities $a_i \in \mathbb{R}$ of a radioactive sample at times $t_i \in \mathbb{R}$. It is known that the sample contains two different radionuclides that decay into non-radioactive isotops. Thus, the sample's total activity is governed by the decay law

$$a(t) = c_1 \exp(-\lambda_1 t) + c_2 \exp(-\lambda_2 t) \ , \quad t \in \mathbb{R} \ . \tag{3}$$

However, the decay rates λ_1, λ_2 and the initial activities c_1, c_2 are not known and are to be estimated from the measurements. This is done by solving a non-linear least squares problem of the form

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{F}(\mathbf{x})\|_2 \ , \tag{4}$$

with a function $\mathbf{F} : \mathbb{R}^m \mapsto \mathbb{R}^n$.

a) **(3 points)** What is $\mathbf{x}$ and $\mathbf{F}$ for the concrete problem of estimating the parameters c_1, c_2, λ_1, λ_2 outlined above?

b) **(4 points)** Give the detailed formulas for one step of the Gauss-Newton method for solving the non-linear least squares problem arising from the current parameter estimation problem.

c) **(8 points)** Write a MATLAB function

```
[c,lambda] = lsq_gauss_newton(t,a)
```

that employs the Gauss-Newton iterations in order to solve the parameter estimation problem for given data t_i and a_i, $i = 1, \ldots, n$, passed in the argument vectors `t` and `a`. The parameters found should be returned in the vectors `c` and `lambda`. What are the values for $c_1, c_2, \lambda_1, \lambda_2$?

To determine an initial guess $\mathbf{x}^{(0)}$ for the Gauss-Newton iteration the fact that one radionuclide decays much more moderate is used. For the initial guess we therefore assume $\lambda_2 = 0$ and $c_2 = 2$. Use the first two measurements (t_1, a_1) and (t_2, a_2) to determine the remaining parameters c_1 and λ_1.

d) **(3 points)** The file `activities.mat` provides data vectors $(t_i)_{i=1}^n$ and $(a_i)_{i=1}^n$. Plot the 2-norm of the error of the Gauss-Newton iterates from `lsq_gauss_newton` versus the number of the iteration step in lin-log scale for errors larger than 10^{-10}. What kind of convergence can you read off the chart?

Problem 5: Method of Heun (12 points)

a) **(3 points)** Let $g : \mathbb{R} \mapsto \mathbb{R}$ be a given Lipschitz continuous function. Convert the scalar third-order ODE

$$\dddot{u} + \sin \dot{u} = g(u) \ , \tag{5}$$

into an equivalent first order ODE.

b) **(3 points)** The method of Heun is a 3-stage explicit Runge-Kutta method described by the Butcher scheme

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{3} & \frac{1}{3} & 0 & \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \tag{6}$$

Write down the formulas for the corresponding discrete evolution for the autonomous ODE $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$.

c) **(3 points)** Implement a MATLAB function

`[t, y] = heun_integrator(odefun, y0, T, N),`

which employs the method of Heun to solve the autonomous initial value problem

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}), \quad \mathbf{y}(t_0) = \mathbf{y}_0 \tag{7}$$

over the interval $[0, T]$ using $N \in \mathbb{N}$ uniform timesteps. The right hand side $\mathbf{f}(\mathbf{y})$ of (7) is passed via the function handle `@(y) odefun(y)`.
The return values should correspond to those of the MATLAB standard integrators.

d) **(3 points)** Write a MATLAB function

`[t, y] = heun_driver(g, y0, T, N),`

that invokes `heun_integrator` from the previous sub-problem to solve initial value problems for the ODE (5).

Hint: By means of the MATLAB script `run_script` you can test `heun_driver` by plotting the solution of

$$\dddot{u} + \sin \dot{u} = -u, \qquad \begin{array}{rcl} u(0) & = & 1 \\ \dot{u}(0) & = & 0 \\ \ddot{u}(0) & = & 0 \end{array} .$$