# cs542g Final Exam
# December 10, 2007

Attempt all questions. Partial marks will be awarded for demonstrating understanding of the relevant material even if you can't fully solve the problem.

**1**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

The thin-plate spline radial basis function in 2D is $r^2 \log r$. This formula is undefined at $r = 0$; assuming the routine for computing $\log r$ has good relative accuracy, is it safe to rely on this formula for $r$ very small but positive? What problem could you run across, and how might you fix it?

**Suggested solution:** Multiplication also has good relative error, so multiplying out $r^2 \log r$ will similarly have good relative error. Using $\epsilon$ as a bound on the relative error, we have

$$\begin{aligned} \textit{computed}(r \cdot r \cdot \log r) &= r(1+\epsilon_1) \cdot r(1+\epsilon_2) \log r (1+\epsilon_3) \\ &\approx r^2 \log r (1+3\epsilon) \end{aligned}$$

The relative error of the result should be about three times larger (worst case) then the relative error of each individual operation. Therefore this is safe, with the one proviso that for extremely small $r$ the quantity $r^2$ might underflow even though $r^2 \log r$ is large enough to be represented accurately: making sure to evaluate it as $r(r \log r)$ makes it entirely safe.

**2**. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Prove how you can get the 2-norm condition number of a general square matrix from its SVD.

**Suggested solution:** Recall the 2-norm condition number of $A$ is $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$. The 2-norm of $A$ can be retrieved from the SVD $A = U\Sigma V^T$ as follows:

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{\|x\|_2=1} \|U\Sigma V^T x\|_2$$

But since $U$ and $V$ are orthogonal matrices, which means they don't change the 2-norm of a vector, substituting $x = Vy$ in the max reduces this to:

$$\|A\|_2 = \max_{\|y\|_2=1} \|\Sigma y\|_2$$

Since $\Sigma$ is diagonal, and assuming the largest singular value comes first, it's easy to see the max is achieved when $y = (1, 0, \dots, 0)^T$. So $\|A\|_2 = \sigma_1$. Similarly, the SVD of $A^{-1}$ is just $V\Sigma^{-1}U^T$ (though now in reverse order) giving $\|A^{-1}\|_2 = 1/\sigma_n$, the reciprocal of the smallest singular value. (Of course, if $A$ is not invertible, this is $\infty$, which will be fine for the condition number). So $\kappa_2(A) = \sigma_1/\sigma_n$.

**3**............................................................

To cut down on the major expense of Newton's method for optimization, namely solving a linear system with the Hessian $H$, you could approximate $H$ as a diagonal matrix $\bar{H}$ with $\bar{H}_{ii} = H_{ii}$ and thus make the solve trivial. How might this impact convergence?

**Suggested solution:** Obviously if $H$ were diagonal to begin with, this won't change anything. More generally it's harder to be sure. Plugging in a model quadratic objective function ($H$ constant) for which true Newton would converge in one step, we see that we are approximating the solution of the linear equation $H\Delta x = -g$ with just the diagonal part of $H$. This is Jacobi iteration, which we have seen may only converge at a linear rate, or not at all. On the other hand, we can expect this to be an improvement over Steepest Descent, where $H$ is approximated as just the identity matrix: if nothing else, the proposed method will usually give a natural step size which Steepest Descent lacks.

**4**............................................................

The following is proposed for solving the second order problem $d^2x/dt^2 = a(x)$:

$$\begin{aligned} v_{n+1} &= v_n + \Delta t\, a(x_n) \\ x_{n+1} &= x_n + \Delta t \frac{v_n + v_{n+1}}{2} \end{aligned}$$

Is this an implicit or an explicit method? How accurate is it? How stable is it?

**Suggested solution:** The velocity step is purely explicit, and while you could argue the position step is implicit since it uses the new velocity, it is basically explicit since this can be reduced to:

$$x_{n+1} = x_n + \Delta t v_n + \frac{1}{2}\Delta t^2 a(x_n)$$

In this form it's eeasy to identify the Taylor series: the local truncation error for velocity is second order, and for position is third order. The global errors should be one less, first and second respectively—you could argue that this method should be labeled as either.

To test stability, linearize the equation: replace $a(x)$ with $-kx$. (From class, we can expect $k$ real and positive for stable dynamics.) Plugging this in gives

$$\begin{aligned} v_{n+1} &= v_n - \Delta t k x_n \\ x_{n+1} &= x_n + \Delta t v_n - \frac{1}{2}\Delta t^2 k x_n \end{aligned}$$

or in matrix-vector form:

$$\begin{pmatrix} v \\ x \end{pmatrix}_{n+1} = \begin{pmatrix} 1 & -\Delta t k \\ \Delta t & 1 - \frac{1}{2}\Delta t^2 k \end{pmatrix} \begin{pmatrix} v \\ x \end{pmatrix}_n$$

Therefore the full solution at time step $n$ is:

$$\begin{pmatrix} v \\ x \end{pmatrix}_n = \begin{pmatrix} 1 & -\Delta t k \\ \Delta t & 1 - \frac{1}{2}\Delta t^2 k \end{pmatrix}^n \begin{pmatrix} v_0 \\ x_0 \end{pmatrix}$$

Stability depends on whether the eigenvalues of this $2 \times 2$ matrix are bounded by 1 in magnitude. However, their product (the determinant of the matrix) is $1 - \frac{1}{2}\Delta t^2 k$, which is always greater than 1, so at least one eigenvalue must have magnitude larger than 1. Therefore the method is unconditionally unstable.

**5. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .**

Consider taking a 2D regular grid, and solving the Poisson equation on it. Write down the simplest 2nd order accurate central finite difference scheme. If you split each grid square into two triangles by adding a diagonal, thus getting a triangle mesh, show that the stiffness matrix from Galerkin FEM with the usual continuous piecewise linear basis functions on that mesh is the same as the finite difference matrix up to a constant factor.

**Suggested solution:** The usual finite difference discretization for $\nabla \cdot \nabla u = f$ is:

$$\frac{u_{i+1,j} + ui, j + 1 - 4u_{i,j} + u_{i,j-1} + u_{i-1,j}}{\Delta x^2} = f_{i,j}$$

ignoring boundary conditions. The stencil connects grid point $(i, j)$ just to its four nearest neighbours.

Define basis functions $\phi_{i,j}(x, y)$ so that $\phi_{i,j}$ is 1 at grid point $(i, j)$ and zero at others, and is linear within each triangle. The Galerkin stiffness matrix has an entry corresponding to the row of grid point $(i, j)$ and column of grid point $(k, l)$ of:

$$A_{(i,j),(k,l)} = \int \nabla\phi_{i,j} \cdot \nabla\phi_{k,l}$$

Just from the support of each basis function, this is zero if $(i, j)$ and $(k, l)$ are not connected by a mesh edge. However, we might have a nonzero offdiagonal when they are connected by a diagonal edge—say $(k, l) = (i + 1, j + 1)$—which is not present in the finite difference scheme.

Observe that since the $\phi$ are linear in each triangle, $\nabla\phi$ is just a constant vector in each triangle. In a triangle where $\phi_{i,j}$ is nonzero, its gradient obviously points towards corner $(i, j)$ and must be perpendicular to the opposite edge (since $\phi_{i,j}$ is a constant zero along that edge, and the gradient of a function is perpendicular to its isocontours). It's easy to see then that if $(i, j)$ and $(k, l)$ are only connected by a diagonal, their gradients are always perpendicular—hence the corresponding term in $A$ is zero.

Therefore the stiffness matrix has the same nonzero pattern as the finite difference scheme. From symmetry, the offdiagonal entries in each row are all equal, just like the finite difference scheme. Finally, if you multiply the stiffness matrix by the vector of all ones, which means integrating $\nabla\phi \cdot \nabla 1 = 0$, we get zero: therefore the sum of each row of $A$ is zero, just like the finite difference scheme. So up to some constant factor which we won't bother computing, the matrices are the same.