Higher-dimensional problems

Numerics in two dimensions

In order to plot a function of the form $u(x,y)$ we need to specify grid points

$$x_\ell \qquad \ell = 1, \ldots, n_t$$

$$y_k \qquad k = 1, \ldots, n_s$$

where we evaluate u.

Notational clash

a two dimensional array is a matrix

$$(a_{k\ell}) = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & \cdots & \\ \vdots & & & \\ a_{n1} & \cdots & & \end{pmatrix}$$

row, column

first entry increases down

second entry increases

whereas for a graph

second entry increases up

first entry increases

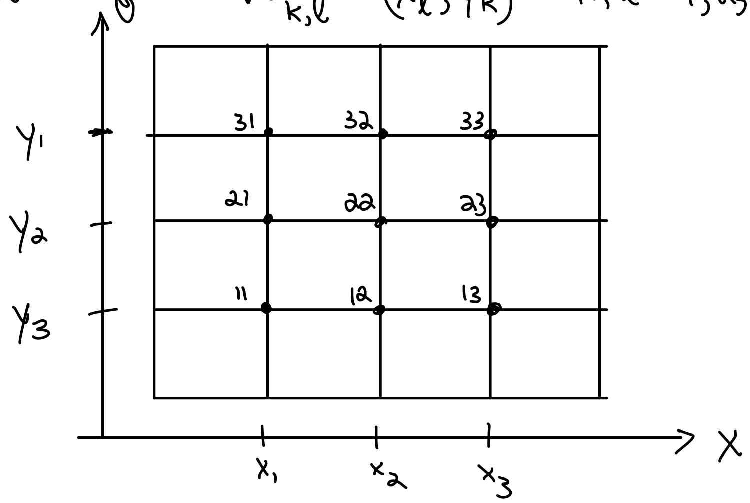
To adjust for this & use the convention

$$(uu_{k\ell}) = u(x_\ell, y_k)$$

Code to graph $u(x,y) = \sin \pi x \, \cos 2\pi y$

```
x = linspace(-1, 1, Nx);
y = linspace(2, 3, Ny);
[xx, yy] = meshgrid(x, y);
uu = sin(pi * xx) .* cos(2*pi*yy);
mesh(xx, yy, uu)
```

Labels for $uu_{k,\ell} = (x_\ell, y_k)$ $k, \ell = 1, 2, 3$ in xx & yy



When working with discretized values it is more convenient to convert matrix u into a vector using the reshape command

$$u = \text{reshape}(uu, N^2, 1)$$

Eg. $uu = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ $\quad \text{reshape}(uu, 4, 1) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$

More generally

$$u = (u_{11}, \dots, u_{N-1\,1}, u_{12}, \dots, u_{N-1,2}, \dots, u_{1N-1} \cdots u_{N-1,N-1})$$

$$f = (f(x_1, y_1), \dots, f(x_1, y_{N-1}), f(x_2, y_1), \dots, f(x_2, y_{N-1}), \dots, f(x_{N-1}, y_{N-1}))$$

Kronecker Product $\quad R = (r_{ij}) \in \mathbb{R}^{a \times b}, \; S \in \mathbb{R}^{k \times j}$

$$\text{kron}(R, S) = \begin{bmatrix} r_{11} S & r_{12} S & \cdots & r_{1b} S \\ r_{21} S & r_{22} S & \cdots & r_{2b} S \\ \vdots & & & \\ r_{a1} S & r_{a2} S & \cdots & r_{ab} S \end{bmatrix}$$

Eg. kron(A, I) vs kron (I, B)

Differentiation in higher dimensions

assume that for $u(x)$ in 1D, $u(x_k) \approx u_k$

there is a 1D differentiation matrix $D \in \mathbb{R}^{N\times N}$

for u on a grid:

$$\begin{pmatrix} u_1' \\ \vdots \\ u_N' \end{pmatrix} \cong D \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}$$

How can we differentiate in the 2D case?

$u_x \cong \mathrm{kron}(D, I)\, u$ where $u = \mathrm{reshape}(uu)$

does the same operation to all of the fixed x points

$u_y \cong \mathrm{kron}(I, D)\, u$

fixed y points are treated the same.

Eg. $\Delta u = u_{xx} + u_{xx}$

$$\Delta u \cong \mathrm{kron}(D^2, I) + \mathrm{kron}(I, D^2)$$

Completeness of two-dimensional families

Theorem

Let $\Omega = (0,a)\times(0,b)$ be a rectangular domain. Assume

$\{\chi_m\}$ is a complete orthogonal set on $L^2(0,a)$

$\{\psi_n\}$ is a complete orthogonal set on $L^2(0,b)$.

Then $\varphi_{m,n}(x,y) = \chi_m(x)\,\psi_n(y)$

is a complete orthogonal set in $L^2((0,a)\times(0,b))$

Proof: suitable application of Fubini's Theorem.

What this means is that you could make complete orthogonal sets out of $\{\cos(k\pi x)\cos(j\pi x)\}$, $\{e^{2\pi i(jx+ky)}\}$ or any mix!

Multidimensional Fourier transforms

Let $\{\varphi_{mn}(x,y)\} = \{\chi_m(x)\,\psi_n(y)\}$

Let $a_x \le x_\ell \le b_x$ are the grid points associated with interpolation via $\{\chi_m\}$ using transform $d\chi t$

Let $a_y \le y_k \le b_y$ are the grid points associated with interpolation by $\{\psi_n\}$ using transform $d\psi t$

Then given $uu_{k\ell} = u(x_\ell, y_k)$, we can find the interpolation coefficients $aa_{k\ell}$ for

$$\varphi(x,y) = \sum_{k=1}^{N_y} \sum_{\ell=1}^{N_x} aa_{k\ell}\, \chi_m(x)\, \psi_n(y)$$

such that this interpolates u:

$$\varphi(x_\ell, y_k) = u(x_\ell, y_k)$$

Use $d\chi t$ on one dim'n of $uu_{k\ell}$

$d\psi t$ on the other dim'n of the result

such as $\boxed{\begin{array}{c}\text{fft2}\\ \text{dct2}\\ \text{dst2}\end{array}}$

Example: Stationary Ohta Kawasaki Equation

$$-\Delta\left(\Delta u + \lambda(u-u^3)\right) - \lambda_\sigma u = 0$$ where $x \in \Omega$

$\Omega = (0,1)^2$ or $(0,1)^3$

with boundary conditions

$$\frac{\partial u}{\partial n}(x) = 0 \quad \text{outward normal on } \partial\Omega$$

and $$\frac{\partial \Delta u}{\partial n} = 0$$ (4th order equation $\Rightarrow$ extra condn)

$$\varphi_{mn}(x_1,x_2) = \cos(m\pi x_1)\cos(n\pi x_2)$$

$$\Delta\varphi_{mn} = -(m^2+n^2)\pi^2\,\varphi_{mn}(x_1,x_2)$$

The strategy is the same as in 1D:

write

$$u(x_1,x_2) = \sum_{n=0}^{N_y-1}\sum_{m=0}^{N_x-1} a_{mn}\,\varphi_{mn}(x)$$

Use the 2-dimensional transform:

Let a_{mn} be a starting guess

Let $uu = \sqrt{N_x}\sqrt{N_y}\,\text{idct}(\text{idct}(a_{mn})')'$ (with padding)

Find $uu - uu.^3$ and unpad to get w

Transform back and divide by $\sqrt{N_x N_y}$

Multiply the terms of a_{nm} by

$-\pi^2(m^2+n^2)$ for each Δ.

Thus we get

$$c2 = \pi(m^2+n^2)\left(-\pi^2(m^2+n^2) + \lambda\, w_{mn}\right) - \lambda\sigma\, a_{mn}$$

reshape c2 to be a long column.

This is when we apply Newton's method.

What about the Jacobian?

We could compute a derivative, but it is also possible to sub in an approximate derivative $e_k = (0 \ldots 0 \overset{k\text{th}}{1} 0 \ldots 0)^t$

h is small (10^{-6} is about right for double precision)

k-column $A(:,k) = \left(F(a+h\cdot e_k) - F(a-h\cdot e_k)\right)/2h$