Visualization of Large Scale Data Sets from Scientific Simulations

W.H. Enright
Department of Computer Science
University of Toronto
Toronto, Canada M5S 3G4

enright@cs.utoronto.ca

- Results from scientific simulations are generally approximate solutions to large systems of Differential Equations. Usually PDEs rather than ODEs.

- Only a very small fraction of the data will ever be viewed by anyone. (Why generate and store the data if nobody will ever see it?)

- Generic examples arise in climate modelling, weather forecasting, and nuclear waste management.

Inherent Difficulties

- Generating the data – New PDE algorithms (HP scientific computation).
- Better mathematical models.
- Storing the data – Data structures and hierarchichal storage devices.
- Transmitting the data – Bandwidth and distributed computing/ collaboration.
- Viewing the Data – Contour plots (2D) and level surfaces (3D). The use of colour, sound and animation.
- Mining the data – searching for patterns, identifying new patterns. Interactive 'steering' of renderer.

Issues to be addressed:

- Accuracy/Resolution

- Cost – both computer time and storage requirement

Related difficulties and Problems:

- Image compression

- Interpolation of scattered unstructured data

We will focus on difficulties involving realistic rendering of accurate ‘coarse’ data.

- A one dimensional example – an ODE.

- Extend this approach to 2D and 3D – PDEs.

- Generic 2D example is the need to generate temperature contours or isobar charts in weather maps. Animation can be used to show evolution over time and 3D (level surfaces) must be used to predict such things as freezing rain or ice storms.

The basic idea of this approach is to use a ‘just-in-time’ strategy for refining the data for rendering.

- Store/transmit accurate coarse data only.

- Generate fine mesh data only when viewing a particular portion of the data. (Request generated and responded to in real time.)

- Use local information only during refinement (ideal for implementation on multiprocessors).

I) A one Dimensional Example

A predator-prey relationship can be modelled by the IVP:

$$y_1^{'} = y_1 - 0.1 y_1 y_2 + 0.02x$$

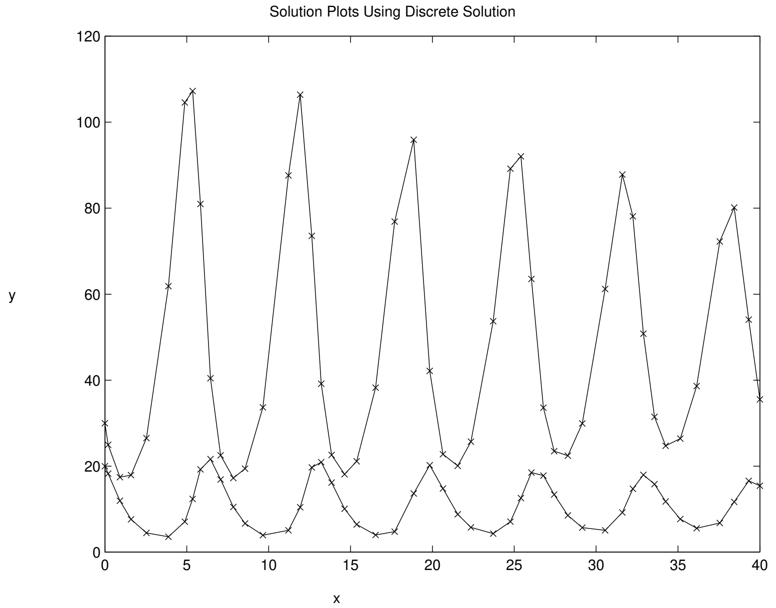
$$y_2^{'} = -y_2 + 0.02 y_1 y_2 + 0.008x$$

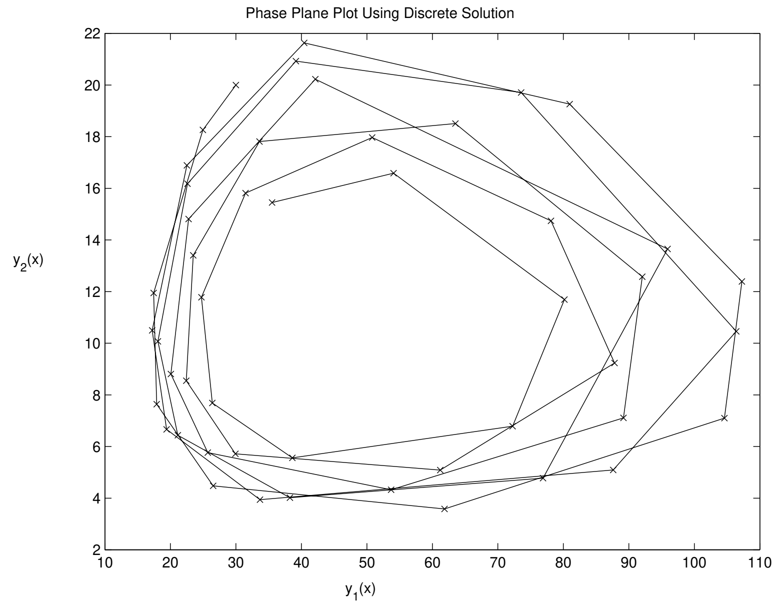
with

$$y_1(0) = 30, \;\; y_2(0) = 20,$$

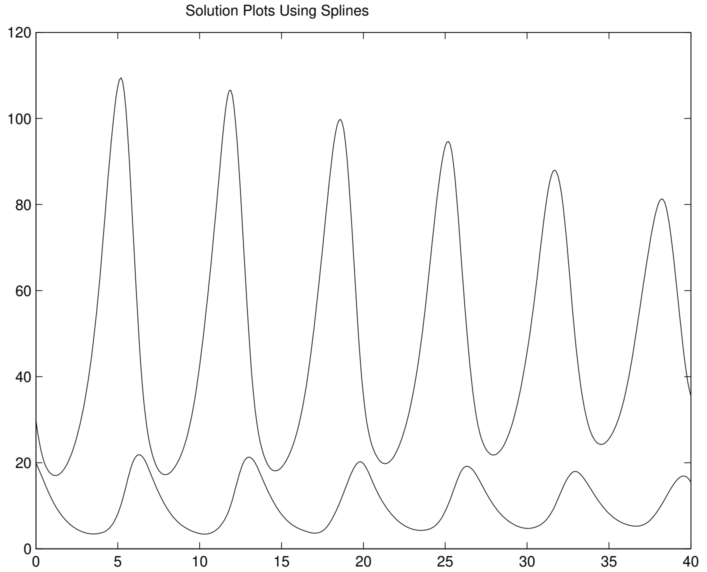
where $y_1(x)$ represents the 'prey' population at time x and $y_2(x)$ represents the 'predator' population at time x. The solution can then be visualized as a standard x/y solution plot or by a 'phase plane' plot. Consider the use of a standard ODE solver such as ode45 of MATLAB.

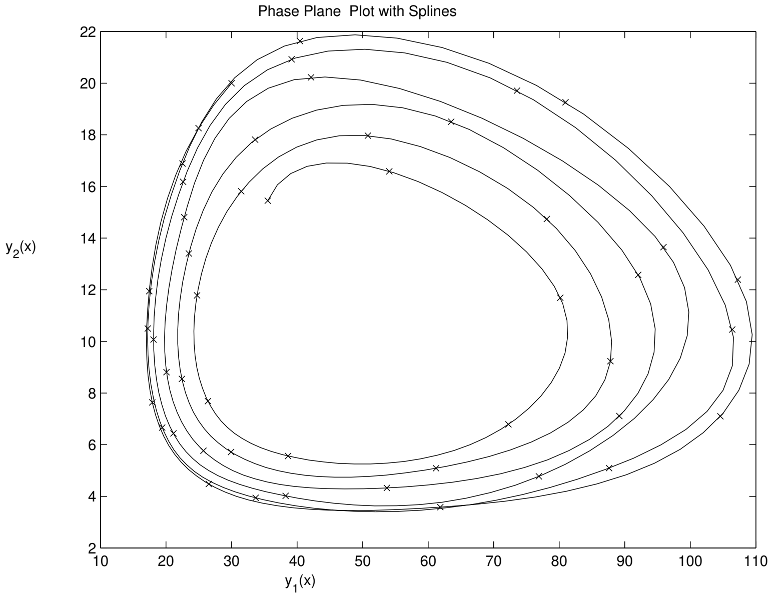
1. Visualizing Solutions with Piecewise Linear interpolation of the Discrete Solution:





2. Visualizing Solutions with Spline interpolation of the Discrete Solution:





3. Visualizing Solutions with new Interpolation of the Discrete Solution: (a 20:1 mesh ratio)

Figure 2. Solution Plots

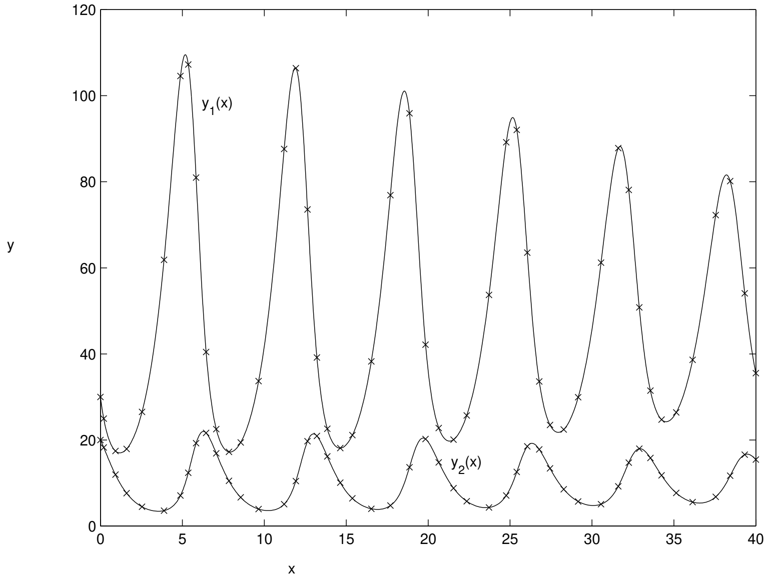
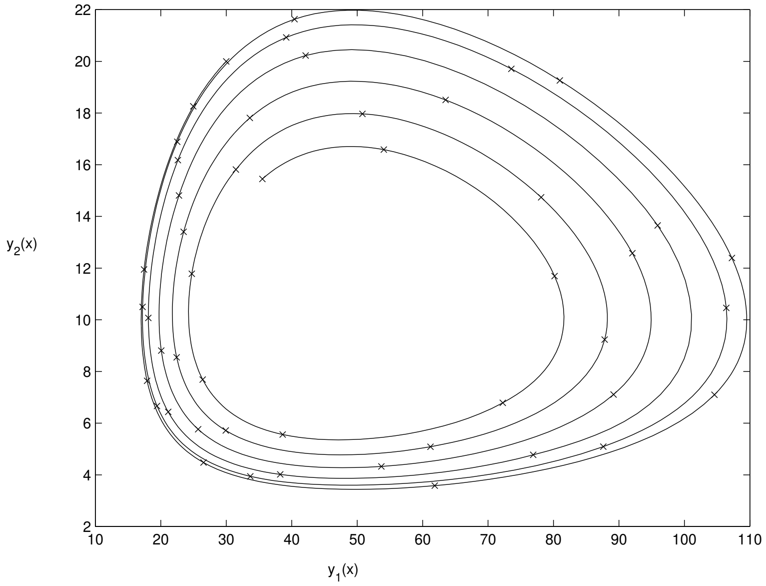


Figure 1. Phase Plane Plot



$\boxed{\text{II) The Generic Approach in one Dimension}}$

Consider the 1^{st}-order system

$$y' = f(t, y).$$

A p^{th}-order, s-stage RK method determines

$$y_i = y_{i-1} + h \sum_{j=1}^{s} \omega_j k_j,$$

where

$$k_j = f(t_{i-1} + hc_j, y_{i-1} + h \sum_{r=1}^{s} a_{jr} k_r).$$

A Continuous extension (CRK) is determined by adding extra stages to obtain an order p approximation for $t \in (t_{i-1}, t_i)$

$$u_i(t) = y_{i-1} + h \sum_{j=1}^{\bar{s}} b_j(\frac{t - t_{i-1}}{h}) k_j,$$

where $b_j(\tau)$ is a polynomial of degree p.

That is, one determines $k_{s+1}, k_{s+2} \dots k_{\bar{s}}$ and polynomials $b_j(\tau)$ to ensure:

$$u_i(t_i) = y_i,$$

$$u_i^{'}(t_i) = f(t_i, y_i),$$

$$u_i(t) = y(t) + O(h^p), \quad t \in [t_{i-1}, t_i].$$

Collectively the $u_i(t)$ define a piecewise polynomial approximation to $y(t)$ that is $O(h^p)$ accurate for $t > t_0$.

$\boxed{\text{III) A Two Dimensional Example}}$

The PDE is from the ELLPACK collection

$u_{xx} + u_{yy} = \cos(\pi y)u - (1 + \sin(\pi x))u_x + f(x, y)$

on the domain

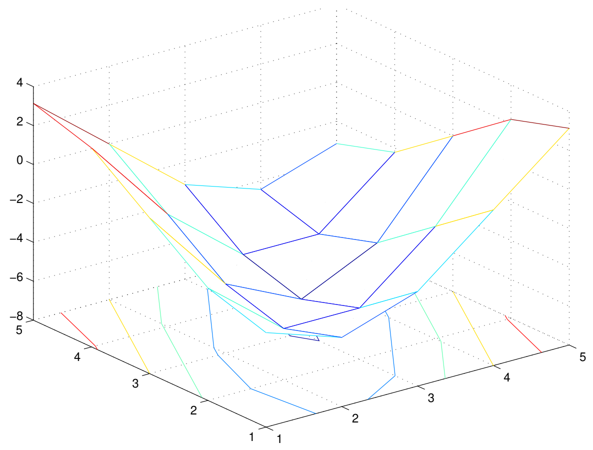
$$0 \leq x \leq 1,\ 0 \leq y \leq 1,$$

with boundary conditions

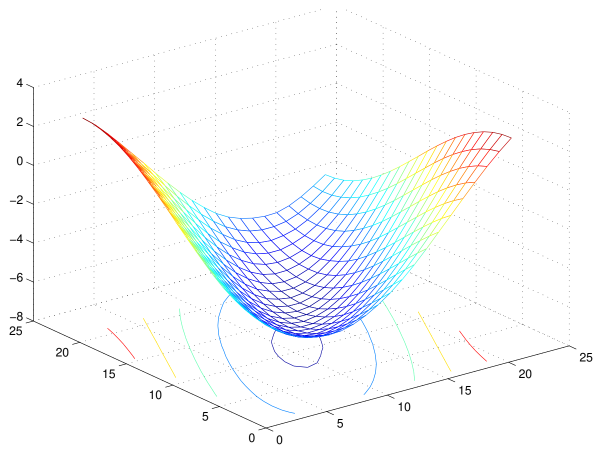
$$u(x, y) = cos(By) + sin\, B(x - y),$$

where $B = \pi$ or $B = 10$.

Consider solving this PDE (with $B = \pi$) on a 4×4 mesh. If we render directly we obtain

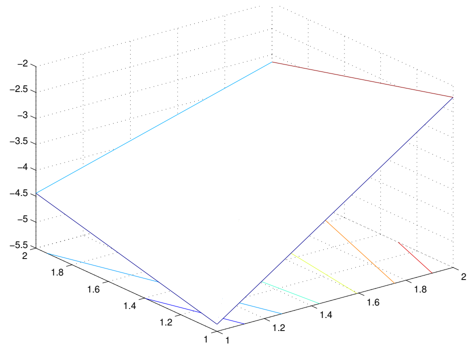


If we interpolate on a fine mesh and render

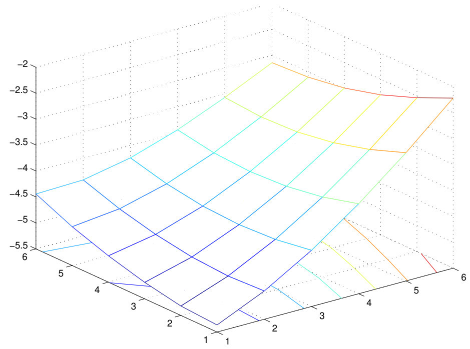


For each mesh element, e, we introduce a bivariate interpolant $\bar{p}_{d,e}(x, y)$ of degree d that interpolates the 'local' meshpoint data and attempts to satisfy the underlying PDE over e.

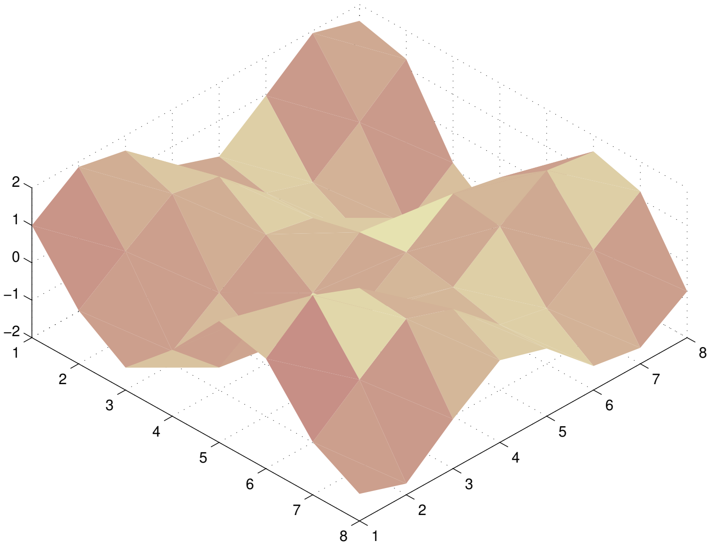
The underlying method produces for e

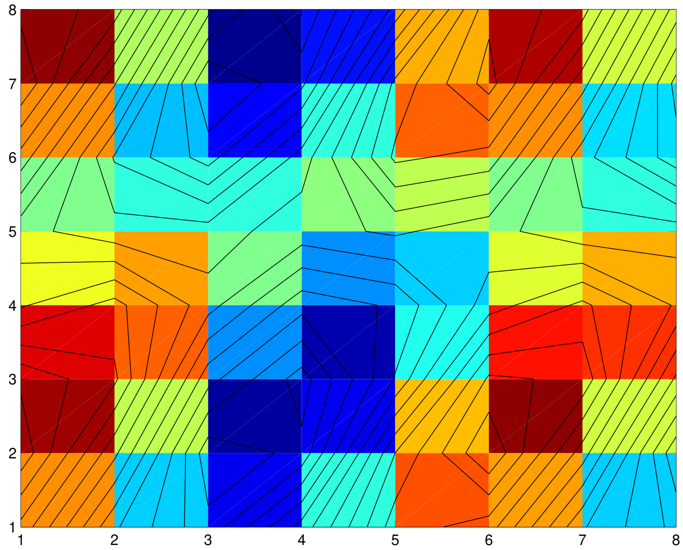


Using $\bar{p}_{3,e}(x, y)$ evaluated on a 5×5 'refined' mesh we obtain

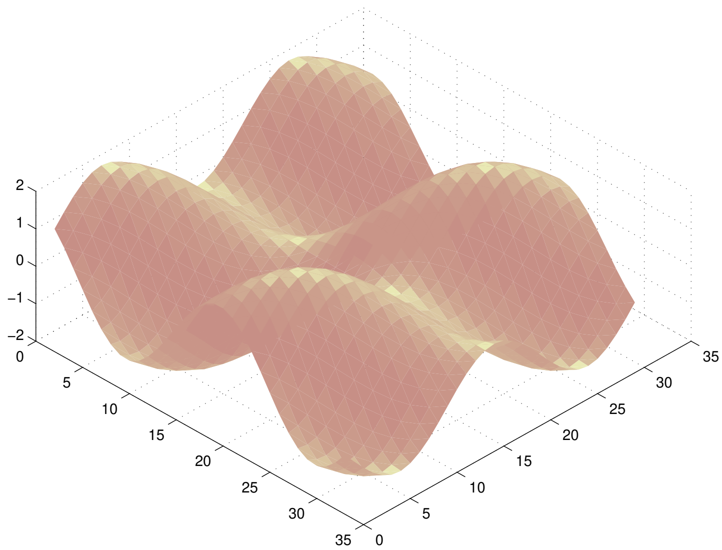


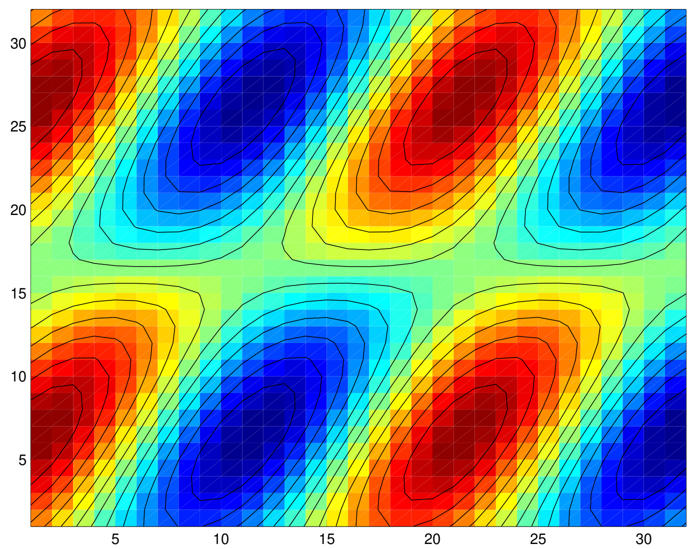
For the PDE defined by $B = 10$, visualizing with 8×8 coarse mesh (Discrete solution only):



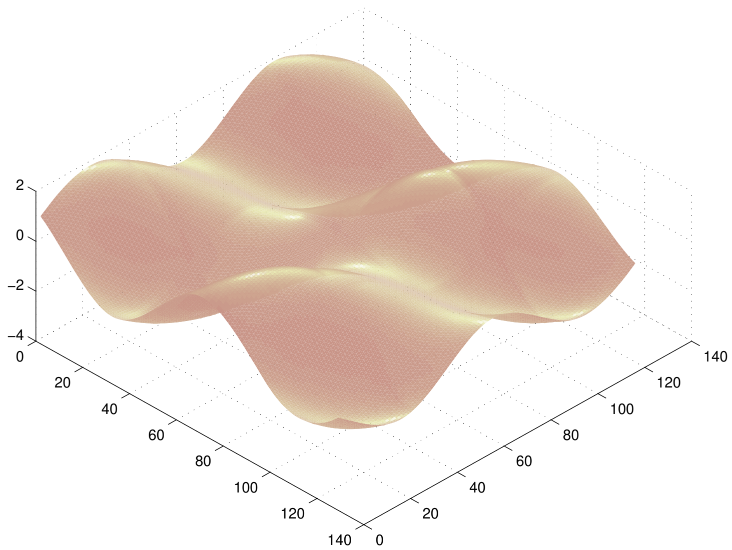


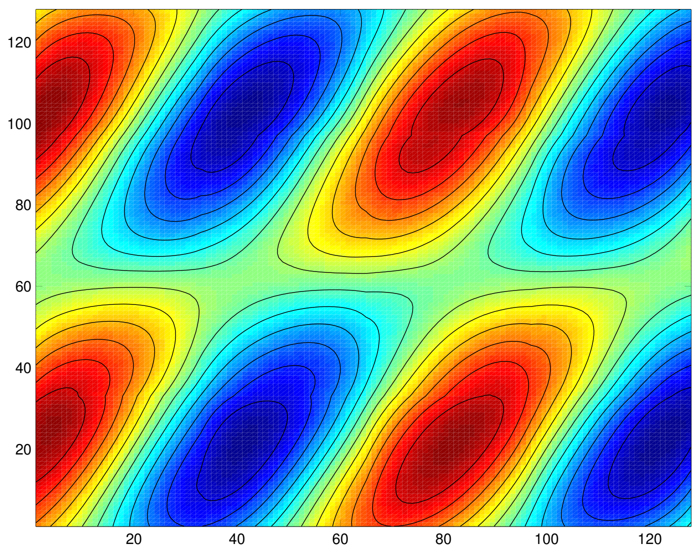
Visualizing with 8×8 coarse mesh and 4×4 fine mesh:



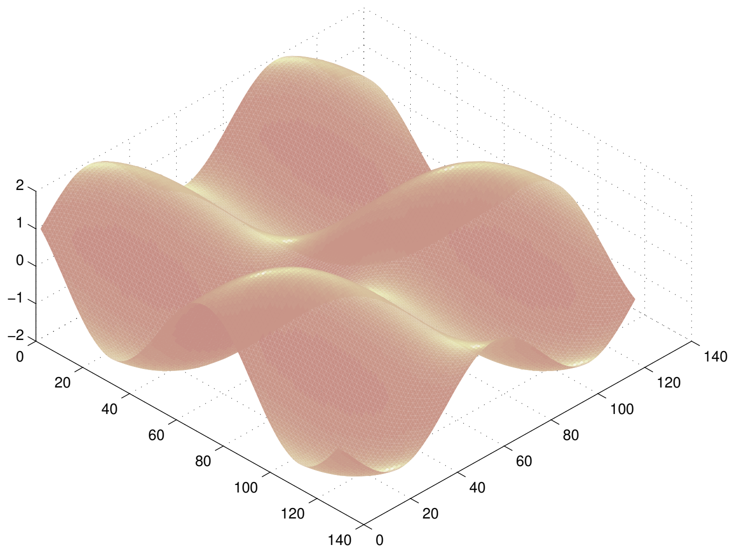


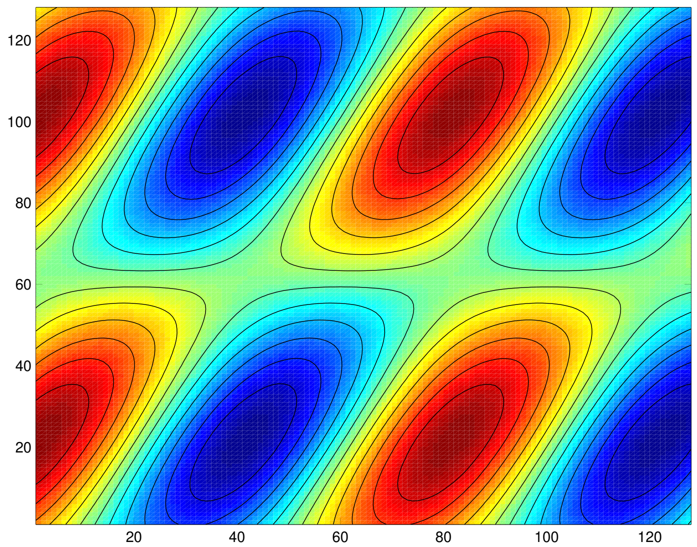
Visualizing with 4×4 coarse mesh and 32×32 fine mesh:





Visualizing with 8×8 coarse mesh and 16×16 fine mesh:





IV) Generic Approach for PDEs

Assume the PDE is a 2D, 2^{nd}-order problem:

$$Lu = g(x, y, u, u_x, u_y), \tag{1}$$

where

$$L = a_1(x, y)u_{xx} + a_2(x, y)u_{yy} + a_3(x, y)u_{xy}.$$

Assume also that an underlying method has generated a discrete approximation to u, u_x and u_y on a grid consisting of rectangular elements. The approximations associated with each meshpoint, $(\bar{x}, \bar{y})$, satisfy

$$\begin{aligned} u(\bar{x}, \bar{y}) - \bar{u} &= O(h^{p_1}), \\ u_x(\bar{x}, \bar{y}) - \bar{u}_x &= O(h^{p_2}), \\ u_y(\bar{x}, \bar{y}) - \bar{u}_y &= O(h^{p_2}), \end{aligned} \tag{2}$$

where h is the maximum mesh spacing in either x or y.

We will also assume that we have $O(h^2)$ accuracy at off-mesh points.

Objectives of this Approach:

For each element, e, determine a bivariate polynomial, $\bar{p}_{d,e}(x, y)$, of degree d that

- interpolates the mesh data (12 constraints)
- almost satisfies (1) at m collocation points
- provides optimum accuracy at off-mesh points

The best we can expect is for $(x, y) \in e$

$$\begin{aligned} u(x,y) - \bar{p}_{d,e}(x,y) &= O(h^p), \\ u_x(x,y) - \frac{\partial \bar{p}_{d,e}(x,y)}{\partial x} &= O(h^{p-1}), \\ u_y(x,y) - \frac{\partial \bar{p}_{d,e}(x,y)}{\partial y} &= O(h^{p-1}), \end{aligned} \tag{3}$$

where $p = min(p_1, p_2 + 1, d + 1)$.

For $(\bar{x}_1, \bar{y}_1)$, one of the meshpoints of e, let $s = (x - \bar{x}_1)/D_1$, $t = (y - \bar{y}_1)/D_2$ and represent $\bar{p}_{d,e}(x,y)$ by

$$\bar{p}_{d,e}(x,y) = \sum_{i=0}^{d} \sum_{j=0}^{d} c_{ij} s^i t^j,$$

where D_1, D_2 are constants that depend on e. The $(d+1)^2$ coefficients, $c = [c_{00}, c_{01}, \cdots c_{dd}]^T$ are determined by specifying $(d+1)^2$ linear constraints. For each of the 4 meshpoints, $(\bar{x}, \bar{y})$, we have 3 interpolation constraints:

$$\begin{aligned}
\sum_{i=0}^{d} \sum_{j=0}^{d} c_{ij} \bar{s}^i \bar{t}^j &= \bar{u}, \\
\sum_{i=0}^{d} \sum_{j=0}^{d} i c_{ij} \bar{s}^{i-1} \bar{t}^j &= D_1 \bar{u}_x, \\
\sum_{i=0}^{d} \sum_{j=0}^{d} j c_{ij} \bar{s}^i \bar{t}^{j-1} &= D_2 \bar{u}_y,
\end{aligned}$$

where $\bar{s} = (\bar{x} - \bar{x}_1)/D_1$, $\bar{t} = (\bar{y} - \bar{y}_1)/D_2$.

Each of the remaining $m = (d+1)^2 - 12$ linear constraints is obtained by applying a collocation condition at a carefully chosen point $(\hat{x}, \hat{y})$ in e. Each partial defining L is linear in the unknowns:

$$\frac{\partial^2 \bar{p}_{d,e}}{\partial x^2} = \frac{1}{D_1^2} \sum_{i=0}^{d} \sum_{j=0}^{d} i(i-1) c_{ij} \hat{s}^{i-2} \hat{t}^{j},$$

$$\frac{\partial^2 \bar{p}_{d,e}}{\partial y^2} = \frac{1}{D_2^2} \sum_{i=0}^{d} \sum_{j=0}^{d} j(j-1) c_{ij} \hat{s}^{i} \hat{t}^{j-2},$$

$$\frac{\partial^2 \bar{p}_{d,e}}{\partial x \partial y} = \frac{1}{D_1 D_2} \sum_{i=0}^{d} \sum_{j=0}^{d} ij c_{ij} \hat{s}^{i-1} \hat{t}^{j-1},$$

where $\hat{s} = (\hat{x} - \bar{x}_1)/D_1$ and $\hat{t} = (\hat{y} - \bar{y}_1)/D_2$.

From (1) we have that $L\bar{p}_{d,e}$ is a linear combination of these 2^{nd}-order partials and therefore the 'collocation' constraint

$$L\bar{p}_{d,e}(\hat{x}, \hat{y}) = g(\hat{x}, \hat{y}, \hat{u}, \hat{u}_x, \hat{u}_y),$$

is a linear equation in the unknowns with a RHS that depends on $\hat{u}$, $\hat{u}_x$ and $\hat{u}_y$.

If we let $h = max(D_1, D_2)$ and order these equations with the interpolating constraints appearing first we obtain

$$Wc = b,$$

where W can be partitioned as

$$W = \left[\frac{W_1}{W_2}\right],$$

and

$$\begin{aligned} b \;=\; & [\bar{u}_1 \cdots \bar{u}_4, D_1\bar{u}_{x1} \cdots D_1\bar{u}_{x4}, D_2\bar{u}_{y1} \cdots \\ & D_2\bar{u}_{y4}, h^2\widehat{g}_1 \cdots h^2\widehat{g}_m]^T. \end{aligned}$$

Recall that our objective is $O(h^p)$ accuracy where $p = min(p_1, p_2 + 1, d + 1)$. To show this we 1st show that if the collocation points are not badly chosen then W^{-1} exists and $\|W^{-1}\|$ is bounded as $h \to 0$. We then show that if $p \leq 4$ we have

$$\|u(x, y) - \bar{p}_{d,e}(x, y)\| = O(h^p).$$

Otherwise $p > 4$ and we must iterate by repeating this procedure R times, where $R \geq p - 3$ to obtain optimal off-mesh accuracy.

With this 'Bootstrapping' Process it can be shown that

$$
\begin{aligned}
\|u(x,y) - \bar{p}_{d,e}^{(R)}(x,y)\| &= O(h^{R+3}), \\
\|u_x(x,y) - \frac{\partial \bar{p}_{d,e}^{(R)}(x,y)}{\partial x}\| &= O(h^{R+2}), \\
\|u_y(x,y) - \frac{\partial \bar{p}_{d,e}^{(R)}(x,y)}{\partial y}\| &= O(h^{R+2}).
\end{aligned}
$$

Note that W is fixed for each iteration and for most PDE methods p is not large and R will rarely exceed 3 .

Approach can be modified and extended in obvious ways:

- 'Non-local' solution values can be used if derivative data not available at meshpoints.

- Can be applied to triangular meshes.

- Can be extended to 3D problems.

- Approach can be applied to first-order and higher-order systems.

- Approach can also be applied to mixed-order systems.

The globally accurate piecewise polynomial $\bar{p}_d(x, y)$ can be effectively used in many situations:

- To generate contour plots or level surfaces.
- To locate discontinuities or identify 'fronts'.
- To perform error estimation and/or mesh refinement.
- To improve heuristics/strategies in the PDE solver.