# Math 6630: Numerical Solutions of Partial Differential Equations
# Finite difference methods for stationary problems

See LeVeque 2007, Chapters 2, 3, 4

Akil Narayan[1]

[1]Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute
University of Utah

January 23, 2023

# Finite difference methods for 1D

Recall: we have discussed finite difference methods for the ODE:

$$
\begin{aligned}
-u''(x) &= f(x), && x \in (0,1) \\
u(0) &= g_0, \\
u(1) &= g_1.
\end{aligned}
$$

The scheme essentially boils down to,

$$
-D_+ D_- u_j = f_j, \qquad\qquad j = 1, \ldots, N,
$$

where,

$$
f_j = f(x_j), \qquad\qquad u_j \approx u(x_j), \qquad\qquad x_j = jh.
$$

We established:

- The scheme amounts to solving an $N \times N$ sparse linear system
- The scheme is second-order convergent

# Finite difference methods for 1D

Recall: we have discussed finite difference methods for the ODE:

$$
\begin{aligned}
-u''(x) &= f(x), & x \in (0,1) \\
u(0) &= g_0, \\
u(1) &= g_1.
\end{aligned}
$$

The scheme essentially boils down to,

$$
-D_+ D_- u_j = f_j, \qquad j = 1, \ldots, \cancel{N}, \; M
$$

where,

$$
f_j = f(x_j), \qquad u_j \approx u(x_j), \qquad x_j = jh.
$$

We established:

- The scheme amounts to solving an $\overset{M}{\cancel{N}} \times \overset{M}{\cancel{N}}$ sparse linear system
- The scheme is second-order convergent

# Partial Differential Equations

The appropriate generalization of our 1D ODE problem is an *elliptic* equation. In 2D, we'll use the notation,

$$u = u(x, y), \qquad \nabla = (\partial_x, \partial_y)^T, \qquad \Delta = \partial_x^2 + \partial_y^2.$$

A fairly general form for a 2D linear elliptic equation is the following:

$$\begin{aligned} -\nabla \cdot (\boldsymbol{\kappa}(x, y)\nabla u) &= f(x, y), & (x, y) &\in (0, 1)^2 \\ u(0, y) = g_0(y),\ u(1, y) &= g_1(y), & y &\in [0, 1] \\ u(x, 0) = h_0(x),\ u(x, 1) &= h_1(x), & x &\in [0, 1], \end{aligned}$$

where $\boldsymbol{\kappa}(x, y)$ is a symmetric matrix that is positive definite everywhere, i.e.,

$$\boldsymbol{v}^T \boldsymbol{\kappa}(x, y)\boldsymbol{v} > 0, \qquad \forall\ (x, y) \in [0, 1]^2,\ \boldsymbol{v} \in \mathbb{R}^2,\ \boldsymbol{v} \neq \boldsymbol{0}.$$

Like the 1D case, this PDE models

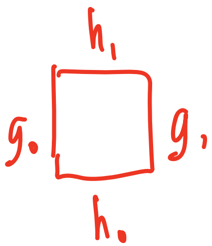
- Spatially-dependent temperature $u$ due to heat diffusion
- $\boldsymbol{\kappa}$ encodes the heat diffusion, allowing heterogeneous, anisotropic heat diffusion.
- This equation also arises in electrostatics, graviational modeling, fluid flow, ....

# Partial Differential Equations

The appropriate generalization of our 1D ODE problem is an *elliptic* equation. In 2D, we'll use the notation,

$$u = u(x,y), \qquad \nabla = (\partial_x, \partial_y)^T, \qquad \Delta = \partial_x^2 + \partial_y^2.$$

A fairly general form for a 2D linear elliptic equation is the following:




$$\begin{aligned}
-\nabla \cdot (\boldsymbol{\kappa}(x,y)\nabla u) &= f(x,y), && (x,y) \in (0,1)^2 \\
u(0,y) = g_0(y),\ u(1,y) &= g_1(y), && y \in [0,1] \\
u(x,0) = h_0(x),\ u(x,1) &= h_1(x), && x \in [0,1],
\end{aligned}$$

where $\boldsymbol{\kappa}(x,y)$ is a symmetric matrix that is positive definite everywhere, i.e.,

$$\boldsymbol{v}^T \boldsymbol{\kappa}(x,y)\boldsymbol{v} > 0, \qquad \forall\ (x,y) \in [0,1]^2,\ \boldsymbol{v} \in \mathbb{R}^2,\ \boldsymbol{v} \neq \mathbf{0}.$$

Like the 1D case, this PDE models

- Spatially-dependent temperature $u$ due to heat diffusion
- $\kappa$ encodes the heat diffusion, allowing heterogeneous, anisotropic heat diffusion.
- This equation also arises in electrostatics, graviational modeling, fluid flow, ....

# Partial Differential Equations

The appropriate generalization of our 1D ODE problem is an *elliptic* equation. In 2D, we'll use the notation,

$$u = u(x,y), \qquad \nabla = (\partial_x, \partial_y)^T, \qquad \Delta = \partial_x^2 + \partial_y^2.$$

A fairly general form for a 2D linear elliptic equation is the following:

$$\begin{aligned}
-\nabla \cdot (\boldsymbol{\kappa}(x,y)\nabla u) &= f(x,y), & (x,y) &\in (0,1)^2 \\
u(0,y) = g_0(y),\ u(1,y) &= g_1(y), & y &\in [0,1] \\
u(x,0) = h_0(x),\ u(x,1) &= h_1(x), & x &\in [0,1],
\end{aligned}$$

where $\boldsymbol{\kappa}(x,y)$ is a symmetric matrix that is positive definite everywhere, i.e.,

$$\boldsymbol{v}^T \boldsymbol{\kappa}(x,y)\boldsymbol{v} > 0, \qquad \forall\ (x,y) \in [0,1]^2,\ \boldsymbol{v} \in \mathbb{R}^2,\ \boldsymbol{v} \neq \mathbf{0}.$$

Like the 1D case, this PDE models

- Spatially-dependent temperature $u$ due to heat diffusion
- $\boldsymbol{\kappa}$ encodes the heat diffusion, allowing heterogeneous, anisotropic heat diffusion.
- This equation also arises in electrostatics, graviational modeling, fluid flow, ....

# Common Specializations

The general elliptic problem is more recognizable with certain simplifications:

If we take $\boldsymbol{\kappa} = \boldsymbol{I}$, then we obtain **Poisson's equation**:

$$-\Delta u = f$$

If we further specialize to $f = 0$, we obtain **Laplace's equation**:

$$-\Delta u = 0.$$

# FD discretization

For simplicity, consider Poisson's equation:

$$
\begin{aligned}
-\Delta u &= f(x,y), & (x,y) &\in (0,1)^2 \\
u(0,y) = g_0(y),\ \ u(1,y) &= g_1(y), & y &\in [0,1] \\
u(x,0) = h_0(x),\ \ u(x,1) &= h_1(x), & x &\in [0,1],
\end{aligned}
$$

We define a uniform, isotropic grid of mesh spacing $h = 1/(M+1)$ over $[0,1]^2$:

$$
u_{i,j} \approx u(x_i, y_j), \qquad x_i = ih, \qquad y_j = jh,
$$

for $i,j = 0,\ldots,M+1$. The unknowns are $u_{i,j}$ for $i,j = 1,\ldots,M$.

An FD discretization proceeds in essentially the same way as before:

$$
\begin{aligned}
u_{xx}(x_i,y_j) &\approx D_+^x D_-^x u_{i,j} = \frac{1}{h^2}\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right), \\
u_{yy}(x_i,y_j) &\approx D_+^y D_-^y u_{i,j} = \frac{1}{h^2}\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right),
\end{aligned}
$$

with local truncation errors,

$$
\begin{aligned}
D_+^x D_-^x u(x_i,y_j) - u_{xx}(x_i,y_j) &\simeq C h^2 u_{xxxx} = \mathcal{O}(h^2), \\
D_+^y D_-^y u(x_i,y_j) - u_{yy}(x_i,y_j) &\simeq C h^2 u_{yyyy} = \mathcal{O}(h^2),
\end{aligned}
$$

hence we expect second-order accuracy with this discretization.

# FD discretization

For simplicity, consider Poisson's equation:

$$
\begin{aligned}
-\Delta u &= f(x,y), && (x,y) \in (0,1)^2 \\
u(0,y) = g_0(y),\ \ u(1,y) &= g_1(y), && y \in [0,1] \\
u(x,0) = h_0(x),\ \ u(x,1) &= h_1(x), && x \in [0,1],
\end{aligned}
$$

We define a uniform, isotropic grid of mesh spacing $h = 1/(M+1)$ over $[0,1]^2$:

$$
u_{i,j} \approx u(x_i, y_j), \qquad x_i = ih, \qquad y_j = jh,
$$

for $i, j = 0, \ldots, M+1$. The unknowns are $u_{i,j}$ for $i, j = 1, \ldots, M$.

An FD discretization proceeds in essentially the same way as before:

$$
\begin{aligned}
u_{xx}(x_i, y_j) &\approx D_+^x D_-^x u_{i,j} = \frac{1}{h^2}\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right), \\
u_{yy}(x_i, y_j) &\approx D_+^y D_-^y u_{i,j} = \frac{1}{h^2}\left(u_{\underset{i,j+1}{\cancel{i+1,j}}} - 2u_{i,j} + u_{\underset{i,j-1}{\cancel{i-1,j}}}\right),
\end{aligned}
$$

with local truncation errors,

$$
\begin{aligned}
D_+^x D_-^x u(x_i, y_j) - u_{xx}(x_i, y_j) &\simeq C h^2 u_{xxxx} = \mathcal{O}(h^2), \\
D_+^y D_-^y u(x_i, y_j) - u_{yy}(x_i, y_j) &\simeq C h^2 u_{yyyy} = \mathcal{O}(h^2),
\end{aligned}
$$

hence we expect second-order accuracy with this discretization.

# The scheme

$$-\int_{\mathbb{R}^d} (\Delta u) u \, d\underline{x} \overset{\text{IBP}}{=} \int_{\mathbb{R}^d} \|\nabla u\|^2 \, d\underline{x}$$

The full scheme is then given by,

$$\begin{array}{ccccc} & -u_{i,j+1} & & & \\ -u_{i-1,j} & +4u_{i,j} & -u_{i+1,j} & = h^2 f_{i,j}, & \qquad i,j = 1,\ldots,M. \\ & -u_{i,j-1} & & & \end{array}$$

with the boundary conditions,

$$\begin{aligned} u_{0,j} &= g_0(y_j), & \qquad u_{\overset{M+1}{\cancel{1}},j} &= g_1(y_j), \\ u_{i,0} &= h_0(x_i), & \qquad u_{i,\underset{M+1}{\cancel{1}}} &= h_1(x_i). \end{aligned}$$

Note that above we approximate $\Delta u$ with grid values on a 5-point *stencil*. Hence we are using a 5-point stencil approximation for the Laplacian.

As one might expect, the above can again be written as a linear system:

$$\boldsymbol{A}\boldsymbol{u} = \widehat{\boldsymbol{f}}, \qquad \boldsymbol{u} = (u_{i,j})_{i,j=1}^{M},$$

where $\widehat{\boldsymbol{f}}$ is a vector depending only on $f$ and the boundary conditions.

# The scheme

The full scheme is then given by,

$$\begin{array}{cccc} & -u_{i,j+1} & & \\ -u_{i-1,j} & +4u_{i,j} & -u_{i+1,j} & = h^2 f_{i,j}, \\ & -u_{i,j-1} & & \end{array} \qquad\qquad i,j = 1,\ldots,M.$$

with the boundary conditions,

$$\begin{aligned} u_{0,j} &= g_0(y_j), & \qquad\qquad u_{1,j} &= g_1(y_j), \\ u_{i,0} &= h_0(x_i), & u_{i,1} &= h_1(x_i). \end{aligned}$$

Note that above we approximate $\Delta u$ with grid values on a 5-point *stencil*. Hence we are using a 5-point stencil approximation for the Laplacian.

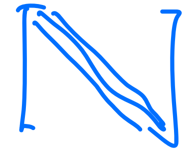
As one might expect, the above can again be written as a linear system:

$$\boldsymbol{A}\boldsymbol{u} = \widehat{\boldsymbol{f}}, \qquad\qquad \boldsymbol{u} = \left(u_{i,j}\right)_{i,j=1}^{M},$$

where $\widehat{\boldsymbol{f}}$ is a vector depending only on $f$ and the boundary conditions.

# Computational considerations in 2D

1D:



$$\boldsymbol{A}\boldsymbol{u} = \widehat{\boldsymbol{f}}, \qquad\qquad \boldsymbol{u} = (u_{i,j})_{i,j=1}^{M},$$

Unlike in 1D:

- $\boldsymbol{A}$ is *not* a tridiagonal (or pentadiagonal) matrix, but is still sparse
- The ordering of the unknowns $(u_{i,j})_{i,j=1}^{M}$ matters a considerable deal in determining the *sparsity pattern* of $\boldsymbol{A}$.
- $\boldsymbol{A}$ is $M^2 \times M^2$, and $\boldsymbol{u}$ contains $M^2$ degrees of freedom – much larger!
- There are no more simple "tricks" to invert $\boldsymbol{A}$ in $\mathcal{O}(M^2)$ time, although iterative methods can solve the problem in $\mathcal{O}(M^2 \log M)$ time.

However, some things are essentially the same:

- The scheme is second-order accurate (convergent) in $h$. (The LTE is second-order, and the scheme is stable.)
- In 1D, scaling $h$ by $1/2$ attained a reduced error scaled by $1/4$. Since scaling $h$ by $1/2$ doubles the degrees of freedom, this is a *superlinear* (quadratic) payoff.
- In 2D, scaling $h$ by $1/2$ again attains a reduced error scaled by $1/4$. But scaling $h$ by $1/2$ *quadruples* the degrees of freedom, so this is only a *linear* payoff.

# Computational considerations in 2D

$$\boldsymbol{A}\boldsymbol{u} = \widehat{\boldsymbol{f}}, \qquad\qquad \boldsymbol{u} = (u_{i,j})_{i,j=1}^{M},$$

Unlike in 1D:

- $\boldsymbol{A}$ is *not* a tridiagonal (or pentadiagonal) matrix, but is still sparse
- The ordering of the unknowns $(u_{i,j})_{i,j=1}^{M}$ matters a considerable deal in determining the *sparsity pattern* of $\boldsymbol{A}$.
- $\boldsymbol{A}$ is $M^2 \times M^2$, and $\boldsymbol{u}$ contains $M^2$ degrees of freedom – much larger!
- There are no more simple "tricks" to invert $\boldsymbol{A}$ in $\mathcal{O}(M^2)$ time, although iterative methods can solve the problem in $\mathcal{O}(M^2 \log M)$ time.

However, some things are essentially the same:

- The scheme is second-order accurate (convergent) in $h$. (The LTE is second-order, and the scheme is stable.)
- In 1D, scaling $h$ by $1/2$ attained a reduced error scaled by $1/4$. Since scaling $h$ by $1/2$ doubles the degrees of freedom, this is a *superlinear* (quadratic) payoff.
- In 2D, scaling $h$ by $1/2$ again attains a reduced error scaled by $1/4$. But scaling $h$ by $1/2$ *quadruples* the degrees of freedom, so this is only a *linear* payoff.

# To higher dimensions

Laplace's equation (indeed, generally any elliptic equation) is essentially the same in an arbitrary number of dimensions $d$:

$$-\Delta u = f, \qquad\qquad \Delta u := \frac{\partial^2 u}{\partial x_1^2} + \cdots + \frac{\partial^2 u}{\partial x_d^2}.$$

As expected, the same FD approach works, discretizing dimension-by-dimension.

The resulting Laplacian stencil has $2d+1$ points – the system matrix $\boldsymbol{A}$ is sparse, with only $2d+1$ non-zero entries per row. ☺

With a uniform, isotropic grid of mesh spacing $h = 1/(M+1)$, there are $M^d \sim (1/h)^d$ degrees of freedom. ☹

Solving the linear system with iterative methods can be accomplished in slightly superlinear time, $\mathcal{O}(dM^d \log M)$ time. ☺

The scheme is still stable, and the LTE is second-order in $h$. ☺

# To higher dimensions

Laplace's equation (indeed, generally any elliptic equation) is essentially the same in an arbitrary number of dimensions $d$:

$$-\Delta u = f, \qquad\qquad \Delta u := \frac{\partial^2 u}{\partial x_1^2} + \cdots + \frac{\partial^2 u}{\partial x_d^2}.$$

As expected, the same FD approach works, discretizing dimension-by-dimension.

The resulting Laplacian stencil has $2d+1$ points – the system matrix $\boldsymbol{A}$ is sparse, with only $2d+1$ non-zero entries per row. ☺

With a uniform, isotropic grid of mesh spacing $h = 1/(M+1)$, there are $M^d \sim (1/h)^d$ degrees of freedom. ☹

Solving the linear system with iterative methods can be accomplished in slightly superlinear time, $\mathcal{O}(dM^d \log M)$ time. ☺

The scheme is still stable, and the LTE is second-order in $h$. ☺

# To higher dimensions

Laplace's equation (indeed, generally any elliptic equation) is essentially the same in an arbitrary number of dimensions $d$:

$$-\Delta u = f, \qquad\qquad \Delta u := \frac{\partial^2 u}{\partial x_1^2} + \cdots + \frac{\partial^2 u}{\partial x_d^2}.$$

As expected, the same FD approach works, discretizing dimension-by-dimension.

The cost vs. accuracy payoff is sublinear if $d \geqslant 3$. ☹

In particular, $h \leftarrow h/2$ requires $2^d$ times more degrees of freedom, with an error reduced to only $2^{-2}$ times the original amount.

More pedantically, the order of convergence, relative to the number of degrees of freedom $N = M^d$, is $2/d$, i.e., the error scales like $N^{-2/d}$.

This exponential attentuation of convergence is one manifestation of the *curse of dimensionality*.

# Delaying the curse of dimensionality

At least in 2D, there is a "trick" that restores second-order convergence *relative to the degrees of freedom*, i.e., has error that is fourth-order in $h$.

The idea is as follows: we know that the standard 5-point stencil Laplacian approximation satisfies,

$$\Delta_5 u_{i,j} = \frac{1}{h^2}\left(-u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} + 4u_{i,j}\right) \simeq \Delta u(x_i, y_j) + Ch^2\left(u_{xxxx} +\right.$$

The LTE term $u_{xxxx} + u_{yyyy}$ is not something we know how to compute without knowledge of $u$, but this expression is similar to the *biharmonic operator*:

$$\Delta^2 := \Delta\Delta u = (\partial_x^2 + \partial_y^2)(\partial_x^2 + \partial_y^2)u = u_{xxxx} + 2u_{xxyy} + u_{yyyy}.$$

The reason this is interesting is that

$$\Delta^2 u = \Delta\Delta u = \Delta f,$$

and *we know* $f$, so in principle can compute $\Delta f$.

I.e., can we "change" the LTE expression to resemble $\Delta^2 u$?

# Delaying the curse of dimensionality

At least in 2D, there is a "trick" that restores second-order convergence *relative to the degrees of freedom*, i.e., has error that is fourth-order in $h$.

The idea is as follows: we know that the standard $5$-point stencil Laplacian approximation satisfies,

$$\Delta_5 u_{i,j} = \frac{1}{h^2}\left(-u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} + 4u_{i,j}\right) \simeq \Delta u(x_i, y_j) + Ch^2\left(u_{xxxx} + \right.$$

$$\simeq \Delta u(x_i, y_j) + Ch^2(u_{xxxx} + u_{yyyy}) + O(h^4)$$

The LTE term $u_{xxxx} + u_{yyyy}$ is not something we know how to compute without knowledge of $u$, but this expression is similar to the *biharmonic operator*:

$$\Delta^2 := \Delta\Delta u = (\partial_x^2 + \partial_y^2)(\partial_x^2 + \partial_y^2)u = u_{xxxx} + 2u_{xxyy} + u_{yyyy}.$$

The reason this is interesting is that

$$\Delta^2 u = \Delta\Delta u = \Delta f,$$

and *we know* $f$, so in principle can compute $\Delta f$.

I.e., can we "change" the LTE expression to resemble $\Delta^2 u$?

# Delaying the curse of dimensionality

At least in 2D, there is a "trick" that restores second-order convergence *relative to the degrees of freedom*, i.e., has error that is fourth-order in $h$.

The idea is as follows: we know that the standard 5-point stencil Laplacian approximation satisfies,

$$\Delta_5 u_{i,j} = \frac{1}{h^2}\left(-u_{i+1,j} - u_{i-1,j} - u_{i,j+1} - u_{i,j-1} + 4u_{i,j}\right) \simeq \Delta u(x_i, y_j) + Ch^2\left(u_{xxxx} +\right.$$

The LTE term $u_{xxxx} + u_{yyyy}$ is not something we know how to compute without knowledge of $u$, but this expression is similar to the *biharmonic operator*:

$$\Delta^2 := \Delta\Delta u = (\partial_x^2 + \partial_y^2)(\partial_x^2 + \partial_y^2)u = u_{xxxx} + 2u_{xxyy} + u_{yyyy}.$$

The reason this is interesting is that

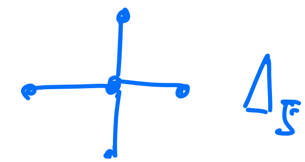
$$\Delta u = f \xrightarrow{\Delta} \qquad \Delta^2 u = \Delta\Delta u = \Delta f,$$

and *we know* $f$, so in principle can compute $\Delta f$.

I.e., can we "change" the LTE expression to resemble $\Delta^2 u$?

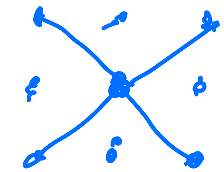
# The 9-point stencil, I

We will attain a biharmonic-like LTE via a combination of two $5$-point stencils. The first stencil is $\Delta_5 u_{i,j}$, that we are already familiar with.




The second stencil is essentially the same, but is "rotated" by $45^\circ$:

$$\widetilde{\Delta}_5 u_{i,j} = \begin{array}{ccc} -u_{i-1,j+1} & & -u_{i+1,j+1} \\ & +4u_{i,j} & \\ -u_{i-1,j-1} & & -u_{i+1,j-1} \end{array} \approx 2h^2 \Delta u(x_i, y_j),$$



The LTE for this approximation similarly contains fourth derivatives, but of a different type.

If we consider a combination of these approximations,

$$\lambda \Delta_5 u_{i,j} + (1-\lambda)\widetilde{\Delta}_5 u_{i,j},$$

and choose $\lambda = 1/3$, then after some (painful) computation, we find,

$$\Delta u(x_i, y_j) + \frac{1}{12} h^2 \left(\Delta^2 u(x_i, y_j)\right) + \mathcal{O}(h^4) \simeq \frac{2}{3}\Delta_5 u_{i,j} + \frac{1}{3}\widetilde{\Delta}_5 u_{i,j}.$$

This results in the $9$-point stencil approximation:

$$\Delta u(x_i, y_j) \approx \Delta_9 u_{i,j} := \frac{1}{6h^2} \begin{pmatrix} -u_{i-1,j+1} & -4u_{i,j+1} & -u_{i+1,j+1} \\ -4u_{i-1,j} & 20u_{i,j} & -4u_{i+1,j} \\ -u_{i-1,j-1} & -4u_{i,j-1} & -u_{i+1,j-1} \end{pmatrix}$$

# The 9-point stencil, I

We will attain a biharmonic-like LTE via a combination of two $5$-point stencils. The first stencil is $\Delta_5 u_{i,j}$, that we are already familiar with.

The second stencil is essentially the same, but is "rotated" by $45^\circ$:

$$\widetilde{\Delta}_5 u_{i,j} = \begin{array}{ccc} -u_{i-1,j+1} & & -u_{i+1,j+1} \\ & +4u_{i,j} & \\ -u_{i-1,j-1} & & -u_{i+1,j-1} \end{array} \approx 2h^2 \Delta u(x_i, y_j),$$

The LTE for this approximation similarly contains fourth derivatives, but of a different type.

If we consider a combination of these approximations,

$$\lambda \Delta_5 u_{i,j} + (1-\lambda)\widetilde{\Delta}_5 u_{i,j},$$

and choose $\lambda = 1/3$, then after some (painful) computation, we find,

$$\Delta u(x_i, y_j) + \frac{1}{12} h^2 \left(\Delta^2 u(x_i, y_j)\right) + \mathcal{O}(h^4) \simeq \frac{2}{3}\Delta_5 u_{i,j} + \frac{1}{3}\widetilde{\Delta}_5 u_{i,j}.$$

This results in the $9$-point stencil approximation:

$$\Delta u(x_i, y_j) \approx \Delta_9 u_{i,j} := \frac{1}{6h^2} \left( \begin{array}{ccc} -u_{i-1,j+1} & -4u_{i,j+1} & -u_{i+1,j+1} \\ -4u_{i-1,j} & 20u_{i,j} & -4u_{i+1,j} \\ -u_{i-1,j-1} & -4u_{i,j-1} & -u_{i+1,j-1} \end{array} \right)$$

# The 9-point stencil, II

What have we accomplished? The LTE for the 9-point approximation is $h^2/12\Delta^2 u + \mathcal{O}(h^4) = h^2/12\Delta f + \mathcal{O}(h^4)$.

$(h^2/12)\Delta^2 u$ $\quad$ $(h^2/12)\Delta f$

For Laplace's equation ($f = 0$), then clearly $\Delta f = 0$, hence, the FD scheme

$$\Delta_9 u_{i,j} = f_{i,j},$$

is *automatically 4th-order accurate in $h$*. Thus, our 9-point stencil achieves 4th order convergence in $h$, or second-order convergence in $M^2 \sim 1/h^2$. I.e., this scheme achieves quadratic accuracy vs cost payoff.

For Poisson's equation ($f \neq 0$), then if we have the ability to compute $F := \Delta f$, then the modified FD scheme,

$$\Delta_9 u_{i,j} = f_{i,j} + \frac{h^2}{12} F_{i,j},$$

will be 4th order accurate in $h$. If $\Delta f$ is not explicitly computable, the same accuracy is achievable via the approximation,

$$\Delta_9 u_{i,j} = f_{i,j} + \frac{h^2}{12} \Delta_5 f_{i,j}.$$

# The 9-point stencil, II

What have we accomplished? The LTE for the 9-point approximation is $h^2/12\Delta^2 u + \mathcal{O}(h^4) = h^2/12\Delta f + \mathcal{O}(h^4)$.

For Laplace's equation ($f = 0$), then clearly $\Delta f = 0$, hence, the FD scheme

$$\Delta_9 u_{i,j} = \cancel{f_{i,j}}, \quad 0$$

is *automatically 4th-order accurate in $h$*. Thus, our $9$-point stencil achieves 4th order convergence in $h$, or second-order convergence in $M^2 \sim 1/h^2$. I.e., this scheme achieves quadratic accuracy vs cost payoff.

For Poisson's equation ($f \neq 0$), then if we have the ability to compute $F := \Delta f$, then the modified FD scheme,

$$\Delta_9 u_{i,j} = f_{i,j} + \frac{h^2}{12} F_{i,j},$$

will be 4th order accurate in $h$. If $\Delta f$ is not explicitly computable, the same accuracy is achievable via the approximation,

$$\Delta_9 u_{i,j} = f_{i,j} + \frac{h^2}{12} \Delta_5 f_{i,j}.$$

## The 9-point stencil, II

What have we accomplished? The LTE for the 9-point approximation is $h^2/12\Delta^2 u + \mathcal{O}(h^4) = h^2/12\Delta f + \mathcal{O}(h^4)$.

For Laplace's equation ($f = 0$), then clearly $\Delta f = 0$, hence, the FD scheme

$$\Delta_9 u_{i,j} = f_{i,j},$$

is *automatically 4th-order accurate in $h$*. Thus, our $9$-point stencil achieves 4th order convergence in $h$, or second-order convergence in $M^2 \sim 1/h^2$. I.e., this scheme achieves quadratic accuracy vs cost payoff.

For Poisson's equation ($f \neq 0$), then if we have the ability to compute $F := \Delta f$, then the modified FD scheme,

$$\Delta_9 u_{i,j} = f_{i,j} + \frac{h^2}{12} F_{i,j},$$

will be 4th order accurate in $h$. If $\Delta f$ is not explicitly computable, the same accuracy is achievable via the approximation,

$$\Delta_9 u_{i,j} = f_{i,j} + \frac{h^2}{12} \Delta_5 f_{i,j}.$$

# Deferred corrections

The previous idea is not really generalizable to other problems, as we must hope that a serendipitous stencil that achieves a particular LTE is identifiable.

The method of *deferred corrections* seeks to make the above idea more practical: for the Poisson problem, first we compute the solution $\widetilde{\boldsymbol{u}}$ to

$$\Delta_5 \widetilde{u}_{i,j} = f_{i,j},$$

and second use $\widetilde{\boldsymbol{u}}$ to compute approximations to the 5-point LTE truncation error

$$\widetilde{\boldsymbol{u}} \xrightarrow{\text{approximate } h^2/12(u_{xxxx}+u_{yyyy})} F_{i,j}$$

Finally, we solve the corrected problem for $\boldsymbol{u}$:

$$\Delta_5 u_{i,j} = f_{i,j} + F_{i,j}$$

With proper construction of $F_{i,j}$, this scheme is again fourth-order accurate in $h$.

# References I

LeVeque, Randall J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM. ISBN: 978-0-89871-783-9.