Solution of Linear Equation Systems

Introduction

- System of algebraic equations: result of discretization process
- Linear or non-linear according to the nature of PDE
- In non-linear case, the discretized equations must be solved by an iterative technique, i.e., *guessing a solution, linearizing the eqns about that solution, improving the solution*, and *repeat*.
- Algebraic eqn. for one CV or grid node in matrix form:

$$A\boldsymbol{\phi} = \boldsymbol{Q} \; . \qquad (5.1)$$

Direct Methods – Gauss Elimination

- Basic method: systematic reduction of large systems of equations to smaller ones

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & A_{n3} & \dots & A_{nn} \end{pmatrix} . \qquad (5.2)$$

SNUTT Seoul National University Towing Tank

Direct Methods – Gauss Elimination

- Technique for eliminating A_{21}, i.e., replacing it with zero.
  - Multiply the 1st row by A_{21}/A_{11}
  - Subtract it from the 2nd row
  - All of the elements in the 2nd row are modified.
  - 2nd element of the forcing vector on RHS is modified.
  - The other elements of the 1st column are treated similarly.
  - By systematically proceeding down, all of the elements below A_{11} are eliminated.
  - When this process is complete, none of the eqns 2, 3, …, n contain ϕ_1.
  - They are a set of n-1 eqns for $\phi_1, \phi_2, \dots \phi_n$.
  - The same procedure is then applied to this smaller set of eqns, i.e., all of the elements below A_{22} in the 2nd column are eliminated.

SNUTT Seoul National University Towing Tank

Gauss Elimination – Cont.

- After carrying out this for columns 1,2,3, …, n-1, the original matrix is replaced by an upper triangular one.

$$U = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ 0 & A_{22} & A_{23} & \dots & A_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & A_{nn} \end{pmatrix} . \qquad (5.3)$$

- It is efficient to store the modified elements in place of the original ones, as the original elements will never be needed again.
- Up to this point, the algorithm is called *forward elimination*.
- Note that the RHS of the eqn, Q_i, are also modified in this process.

SNUTT Seoul National University Towing Tank

Gauss Elimination – Cont.

- The last eqn contains only one variable, ϕ_n

$$\phi_n = \frac{Q_n}{A_{nn}} . \qquad (5.4)$$

- Proceeding upward, the i-th eqn yields ϕ_i – back substitution

$$\phi_i = \frac{Q_i - \sum_{k=i+1}^{n} A_{ik}\phi_k}{A_{ii}} . \qquad (5.5)$$

- For large *n*, the number of operations required to solve a linear system of *n* eqns by Gauss elimination is proportional to $n^3/3$. $\rightarrow$ Expensive
- For large systems that are not sparse, Gauss elimination is susceptible to accumulation of errors.
- Gauss elimination does not vectorize or parallelize well and is rarely used without modification in CFD problems.

SNUTT Seoul National University Towing Tank

Direct Methods – LU Decomposition

- A variation on Gauss elimination
  - Any matrix ***A*** can be factored into the product of lower (***L***) and upper (***U***) triangular matrices:

$$A = LU \ . \qquad (5.6)$$

  - Forward elimination can be carried out in a more formal manner by multiplying ***A*** by a lower triangular matrix.
  - Requirement: the diagonal elements of ***L***, L_{ii}, all be unity.

LU Decomposition – Cont.

- The factorization is easily constructed.
  - ***U*** is precisely the one produced by the forward phase of Gauss elimination.
  - The elements of ***L*** are the multiplicative factors (A_{ji}/A_{ii}) used in the elimination process.
  - This allows the factorization to be constructed by a minor modification of Gauss elimination.
  - The elements of ***L*** and ***U*** can be stored where the elements of ***A*** were.

LU Decomposition – Cont.

- The solution of Eq. (5.1) in two stages:

$$U\phi = \boldsymbol{Y} \ , \tag{5.7}$$

$$L\boldsymbol{Y} = \boldsymbol{Q} \ . \tag{5.8}$$

  - Once Eq. (5.8) has been solved for ***Y***, Eq. (5.7) can be solved for ϕ.
- The advantage of LU factorization over Gauss elimination is that the factorization can be performed without knowing the vector ***Q***.

Direct Methods – Tridiagonal Systems

- When 1D ODE eqns are finite differenced with CDS approximation, the resulting algebraic eqns have a simple structure.

$$A_W^i \phi_{i-1} + A_P^i \phi_i + A_E^i \phi_{i+1} = Q_i \ . \tag{5.9}$$

  - Tridiagonal matrix: non-zero terms only on main diagonal (A_P) and the diagonals immediately above and below it (A_E and A_W). $\rightarrow$ The elements are best stored as 3 nx1 arrays.
  - Gauss elimination is easy: only one element needs to be eliminated from each row during the forward elimination.

$$\bar{A}_P^i = A_P^i - \frac{A_W^i A_E^{i-1}}{\bar{A}_P^{i-1}} \ , \tag{5.10}$$

Tridiagonal Systems – Cont.

- The forcing term is also modified:

$$Q_i^* = Q_i - \frac{A_W^i Q_{i-1}^*}{A_P^{i-1}} . \quad (5.11)$$

$$\phi_i = \frac{Q_i^* - A_E^i \phi_{i+1}}{A_P^i} . \quad (5.12)$$

- Thomas algorithm or Tridiagonal matrix algorithm (TDMA).
- The number of operations is proportional to *n*.

Direct Methods – Cyclic Reduction

- Matrix is not only tridiagonal but all of the elements on each diagonal are identical. → Cyclic reduction.
  - Suppose that in Eq. (5.9), A_W^i, A_P^i and A_E^i are independent of *i*.
  - For even *i*, multiply row *i-1* by A_W/A_P and subtract it from row *i*, then multiplay row *i+1* by A_E/A_P and subtract it from row *i*. → Eliminates the elements to the immediate left and right of the main diagonal in the *even* numbered rows, but replaces the zero element 2 columns to the left by $-A_W^2/A_P$ and the zero element 2 columns to the right by $-A_E^2/A_P$. The diagonal element becomes $A_P - 2A_W A_E/A_P$.
  - Because the elements in every even row are the same, the calculation of the new elements needs to be done only once.

Cyclic Reduction – Cont.

- Cost of this method is proportional to $\log_2 n$.
- Cyclic reduction provides the basis for methods of solving elliptic eqns such as Laplace and Poisson eqns directly, i.e., non-iteratively.

Iterative Methods – Basic Concept

- Any system of eqns can be solved by Gauss elimination or LU decomposition.
- The triangular factors of sparse matrices are not sparse, so the cost of these methods is quite high.
- The discretization error is usually much larger than the accuracy of the computer arithmetic, so there is no reason to solve the system that accurately.
- Solution to somewhat more accuracy than that of the discretization scheme suffices.

Why iterative methods?

Basic Concept – Cont.

- In an iterative method, one guesses a solution, and uses the eqn to systematically improve it -> cheaper than direct methods.
  - Consider Eq. (5.1). After *n* iterations, we have an approximate solution ϕ^n which does not satisfy these eqns exactly. Instead there is a non-zero residual ρ^n:

$$A\phi^n = Q - \rho^n \, . \tag{5.13}$$

  - Subtract this from Eq. (5.1). Definition of iteration error

$$\epsilon^n = \phi - \phi^n \, , \tag{5.14}$$

$$\xrightarrow{} A\left(\phi - \phi^n\right) = Q - \left(Q - \rho^n\right)$$

$$A\epsilon^n = \rho^n \, . \tag{5.15}$$

  - The purpose of the iteration procedure is to drive the residuals to zero, and in the process ε also becomes zero.

SNUTT Seoul National University Towing Tank

Basic Concept – Cont.

- Consider a general iterative scheme for a linear system

$$M\phi^{n+1} = N\phi^n + B \, . \tag{5.16}$$

  - At convergence, since $\phi^{n+1} = \phi^n = \phi$.

$$A = M - N \quad \text{and} \quad B = Q \, , \tag{5.17}$$

$$\xleftarrow{} A\left(\phi^n + \rho^n\right) = M\phi^{n+1} - N\phi^n = Q$$

  - or more generally

$$PA = M - N \quad \text{and} \quad B = PQ \, , \tag{5.18}$$

  - where ***P*** is a non-singular pre-conditioning matrix

SNUTT Seoul National University Towing Tank

Basic Concept – Cont.

- An alternative version
  - Subtract $M\phi^n$ from Eq. (5.16)

$$M(\phi^{n+1} - \phi^n) = \boldsymbol{B} - (M - N)\phi^n \quad \text{or} \quad M\delta^n = \rho^n \,, \tag{5.19}$$

  - Where the correction $\delta^n = \phi^{n+1} - \phi^n$
- For an iterative method to be effective, solving Eq. (5.16) must be cheap and the method must converge rapidly.
  - ***A*** is sparse → ***N*** is sparse → $N\phi^n$ is simple, i.e., $N\phi$ computation is easy.
  - ***M*** must be easily inverted, i.e., diagonal, tridiagonal, triangular, …, i.e., the solution of system is easy.
  - For rapid convergence, ***M*** should be a good approximation to ***A***, making $\boldsymbol{N}\phi$ small.

SNUTT
Seoul National University Towing Tank

Iterative Methods - Convergence

- What determines the convergence rate and how to improve it?
  - Because $\phi^{n+1} = \phi^n = \phi$, at convergence, the converged solution obeys

$$M\phi = N\phi + \boldsymbol{B} \,. \tag{5.20}$$

  - Subtracting this from Eq. (5.16) and using Eq. (5.14)

$$M\epsilon^{n+1} = N\epsilon^n \tag{5.21}$$

or

$$\epsilon^{n+1} = M^{-1}N\epsilon^n \,. \tag{5.22}$$

  - The iterative method converges if $\lim_{n\to\infty} \epsilon^n = 0$.

SNUTT
Seoul National University Towing Tank

Convergence – Cont.

- Now that $\epsilon^{n+1} = M^{-1}N\epsilon^n$, let's consider the eigenvalues λ_k and eigenvectors ψ^k of the iteration matrix $M^{-1}N$

  $$M^{-1}N\psi^k = \lambda_k\psi^k \ , \quad k = 1, \ldots, K \ , \qquad (5.23)$$

  where K is # of eqns (grid points).
- Now assume that the eigenvectors form the vector space if all n-components vectors, then the initial error may be expressed

  $$\epsilon^0 = \sum_{k=1}^{K} a_k\psi^k \ , \qquad (5.24)$$

  where a_k is a constant.
- Then the iterative procedure Eq. (5.22) yields

  $$\epsilon^1 = M^{-1}N\epsilon^0 = M^{-1}N\sum_{k=1}^{K} a_k\psi^k = \sum_{k=1}^{K} a_k\boxed{\lambda_k\psi^k} \qquad (5.25)$$

SNUTT Seoul National University Towing Tank

Convergence – Cont.

- By induction

  $$\epsilon^n = \sum_{k=1}^{K} a_k(\lambda_k)^n\psi^k \ . \qquad (5.26)$$

- For ϵ^n to become zero when *n* is large, the necessary and sufficient condition: all of the eigenvalues < 1.
- The spectral radius, i.e., the largest eigenvalue, must be less than one.
- In fact, after some iterations, the terms with small eigenvalues in Eq. (5.26) become very small and only the terms with the largest eigenvalue remains:

  $$\epsilon^n \sim a_1(\lambda_1)^n\psi^1 \ . \qquad (5.27)$$

SNUTT Seoul National University Towing Tank

Convergence – Cont.

- If convergence is defined as the reduction of the iteration error below some tolerance δ,

$$a_1(\lambda_1)^n \approx \delta \, . \tag{5.28}$$

- An expression for the required # of iterations

$$n \approx \frac{\ln\left(\frac{\delta}{a_1}\right)}{\ln \lambda_1} \, . \tag{5.29}$$

- Simple example: case of a single equation

$$ax = b \tag{5.30}$$

We use the iteration method

$$mx^{p+1} = nx^p + b \, . \tag{5.31}$$

Then the error obeys, as Eq. (5.22)

$$\epsilon^{p+1} = \frac{n}{m}\epsilon^p \, . \tag{5.32}$$

SNUTT Seoul National University Towing Tank

Convergence – Cont.

- The error is reduced quickly if *n/m* is small, i.e., $m \approx a$.
- The more closely ***M*** approximates ***A***, the more rapid the convergence.

SNUTT Seoul National University Towing Tank

Iterative Methods – Some Basic Methods

- Jacobi method
  - Simplest - ***M*** is a diagonal matrix whose elements are the diagonal elements of ***A***

$$\phi_P^{n+1} = \frac{Q_P - A_S\phi_S^n - A_W\phi_W^n - A_N\phi_N^n - A_E\phi_E^n}{A_P} . \qquad (5.33)$$

  - Requires (cell #)^2 operations in 1 direction.
- Gauss-Seidel method
  - $\phi_P^{n+1} = \frac{Q_P - A_S\phi_S^{n+1} - A_W\phi_W^{n+1} - A_N\phi_N^n - A_E\phi_E^n}{A_P}$
  - A special case of the SOR method

Some Basic Methods – Cont.

- Successive over-relaxation (SOR) method
  - Accelerated version of GS method

$$\phi_P^{n+1} = \omega \frac{Q_P - A_S\phi_S^{n+1} - A_W\phi_W^{n+1} - A_N\phi_N^n - A_E\phi_E^n}{A_P} + (1-\omega)\phi_P^n \ ,(5.34)$$

  - ω is the over-relaxation factor
  - HW #6

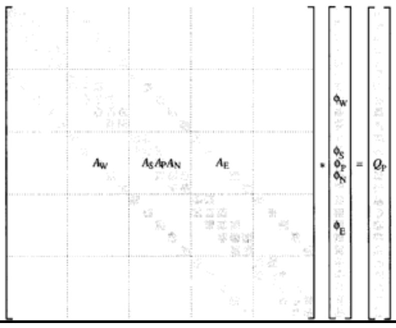
Iterative Methods – Incomplete LU Decomposition

- Use an approximate ***LU*** factorization of ***A*** as the iteration matrix ***M***

$$M = LU = A + N \,, \tag{5.35}$$

  where ***L*** and ***U*** are both sparse and ***N*** is small.

- Strongly implicit procedure (SIP)
  - For 5-point computational molecule



SNUTT Seoul National University Towing Tank

Incomplete LU Decomposition – Cont.

- ***L*** and ***U*** matrices have non-zero elements only on diagonals on which ***A*** has non-zero elements.
- The product of these 2 matrices produce extra 2 diagonals on the nodes NW and SE.
- For matrices below

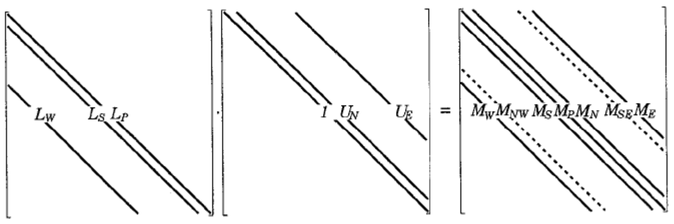


Fig. 5.1. Schematic presentation of the matrices L and U and the product matrix M; diagonals of M not found in A are shown by dashed lines

SNUTT Seoul National University Towing Tank

Incomplete LU Decomposition – Cont.

$$
\begin{aligned}
M_{\mathrm{W}}^l &= L_{\mathrm{W}}^l \\
M_{\mathrm{NW}}^l &= L_{\mathrm{W}}^l U_{\mathrm{N}}^{l-N_j} \\
M_{\mathrm{S}}^l &= L_{\mathrm{S}}^l \\
M_{\mathrm{P}}^l &= L_{\mathrm{W}}^l U_{\mathrm{E}}^{l-N_j} + L_{\mathrm{S}}^l U_{\mathrm{N}}^{l-1} + L_{\mathrm{P}}^l \\
M_{\mathrm{N}}^l &= U_{\mathrm{N}}^l L_{\mathrm{P}}^l \\
M_{\mathrm{SE}}^l &= L_{\mathrm{S}}^l U_{\mathrm{E}}^{l-1} \\
M_{\mathrm{E}}^l &= U_{\mathrm{E}}^l L_{\mathrm{P}}^l
\end{aligned}
\qquad (5.36)
$$

- Note that *l* is the 1-D storage location for grid indices *(i, j)*
- Select ***L*** and ***U***, such that ***M*** is as good an approximation to ***A*** as possible.

SNUTT Seoul National University Towing Tank

Incomplete LU Decomposition – Cont.

Stone's suggestion for acceleration

- Consider the vector ***Mϕ***

$$
(M\phi)_{\mathrm{P}} = M_{\mathrm{P}}\phi_{\mathrm{P}} + M_{\mathrm{S}}\phi_{\mathrm{S}} + M_{\mathrm{N}}\phi_{\mathrm{N}} + M_{\mathrm{E}}\phi_{\mathrm{E}} + M_{\mathrm{W}}\phi_{\mathrm{W}} + M_{\mathrm{NW}}\phi_{\mathrm{NW}} + M_{\mathrm{SE}}\phi_{\mathrm{SE}} \,. \qquad (5.37)
$$

- ***N*** must contain the 2 extra diagonals and we want to choose the elements on the remaining diagonals so that $N\phi \approx \mathbf{0}$ or

$$
N_{\mathrm{P}}\phi_{\mathrm{P}} + N_{\mathrm{N}}\phi_{\mathrm{N}} + N_{\mathrm{S}}\phi_{\mathrm{S}} + N_{\mathrm{E}}\phi_{\mathrm{E}} + N_{\mathrm{W}}\phi_{\mathrm{W}} + M_{\mathrm{NW}}\phi_{\mathrm{NW}} + M_{\mathrm{SE}}\phi_{\mathrm{SE}} \approx 0 \,. \qquad (5.38)
$$

- In other words, Eq. (5.38) reduces to

$$
M_{\mathrm{NW}}(\phi_{\mathrm{NW}} - \phi_{\mathrm{NW}}^*) + M_{\mathrm{SE}}(\phi_{\mathrm{SE}} - \phi_{\mathrm{SE}}^*) \approx 0 \,, \qquad (5.39)
$$

  where ϕ_{NW}^* and ϕ_{SE}^* are approximations to ϕ_{NW} and ϕ_{SE}

- Stone's idea ← Solution should be smooth, because elliptic equation & α < 1 for stability reasons

$$
\begin{aligned}
\phi_{\mathrm{NW}}^* &\approx \alpha(\phi_{\mathrm{W}} + \phi_{\mathrm{N}} - \phi_{\mathrm{P}}) \\
\phi_{\mathrm{SE}}^* &\approx \alpha(\phi_{\mathrm{S}} + \phi_{\mathrm{E}} - \phi_{\mathrm{P}})
\end{aligned}
\qquad (5.40)
$$

SNUTT Seoul National University Towing Tank

Incomplete LU Decomposition – Cont.

- Substitute Eq. (5.40) into Eq. (5.39), then the result is equated to Eq. (5.38) → Obtain elements of ***N*** as linear combinations of M_{NW} and M_{SE}.
- The elements of ***M***, Eq. (5.36), can now be set equal to the sum of elements of ***A*** and ***N***.

$$
\begin{aligned}
L_W^l &= A_W^l / (1 + \alpha U_N^{l-N_j}) \\
L_S^l &= A_S^l / (1 + \alpha U_E^{l-1}) \\
L_P^l &= A_P^l + \alpha (L_W^l U_N^{l-N_j} + L_S^l U_E^{l-1}) - L_W^l U_E^{l-N_j} - L_S^l U_N^{l-1} \qquad (5.41) \\
U_N^l &= (A_N^l - \alpha L_W^l U_N^{l-N_j}) / L_P^l \\
U_E^l &= (A_E^l - \alpha L_S^l U_E^{l-1}) / L_P^l
\end{aligned}
$$

Iterative Methods – ADI and Other Splitting Methods

- A common method of solving elliptic problems
  - Add a term containing the 1st time derivative to the equation
  - Solve the resulting parabolic problem until steady state
- For stability, methods implicit in time is required for parabolic equations → Solution of elliptic problem at each time step
  - Cost can be reduced by using the alternating direction method (ADI)

ADI & Other Splitting Methods – Cont.

- Example: 2D Laplace eqn
  - Add a time derivative → convert it to 2D heat equation

$$\frac{\partial\phi}{\partial t} = \Gamma\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right) . \tag{5.46}$$

  - Discretize using the trapezoid rule (Crank-Nicolson) in time and central difference in space

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} = \frac{\Gamma}{2}\left[\left(\frac{\delta^2\phi^n}{\delta x^2} + \frac{\delta^2\phi^n}{\delta y^2}\right) + \left(\frac{\delta^2\phi^{n+1}}{\delta x^2} + \frac{\delta^2\phi^{n+1}}{\delta y^2}\right)\right] , \tag{5.47}$$

where

$$\left(\frac{\delta^2\phi}{\delta x^2}\right)_{i,j} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta x)^2} ,$$

$$\left(\frac{\delta^2\phi}{\delta y^2}\right)_{i,j} = \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{(\Delta y)^2}$$

ADI & Other Splitting Methods – Cont.

- Rearranging Eq. (5.47), at time step *n+1*

$$\left(1 - \frac{\Gamma\Delta t}{2}\frac{\delta^2}{\delta x^2}\right)\left(1 - \frac{\Gamma\Delta t}{2}\frac{\delta^2}{\delta y^2}\right)\phi^{n+1} =$$
$$\left(1 + \frac{\Gamma\Delta t}{2}\frac{\delta^2}{\delta x^2}\right)\left(1 + \frac{\Gamma\Delta t}{2}\frac{\delta^2}{\delta y^2}\right)\phi^n - \tag{5.48}$$
$$\frac{(\Gamma\Delta t)^2}{4}\frac{\delta^2}{\delta x^2}\left[\frac{\delta^2(\phi^{n+1} - \phi^n)}{\delta y^2}\right] .$$

- As $\phi^{n+1} - \phi^n \approx \Delta t\,\partial\phi/\partial t$ the last term is proportional to $(\Delta t)^3$ for small Δt . Since the FD approximation is of 2nd order, for small Δt, the last term is small compared to the discretizaion error and may be *neglected*.

ADI & Other Splitting Methods – Cont.

- The remaining eqn can be factored into 2 simpler eqns

$$\left(1 - \frac{\Gamma \Delta t}{2} \frac{\delta^2}{\delta x^2}\right) \phi^* = \left(1 + \frac{\Gamma \Delta t}{2} \frac{\delta^2}{\delta y^2}\right) \phi^n \, , \tag{5.49}$$

$$\left(1 - \frac{\Gamma \Delta t}{2} \frac{\delta^2}{\delta y^2}\right) \phi^{n+1} = \left(1 + \frac{\Gamma \Delta t}{2} \frac{\delta^2}{\delta x^2}\right) \phi^* \, . \tag{5.50}$$

- Each is a set of tridiagonal eqns and solved with TDMA → No iteration and much cheaper than solving Eq. (5.47)
- Either Eq. (5.49) or (5.50) is 1st order accurate and conditionally stable, but the combined method is 2nd order accurate and unconditionally stable.
- The family of methods based on these ideas: ***splitting*** or ***approximate factorization***
- Neglect of 3rd order term (essential to the factorization) is justified only when the time step is small (usually true for pressure or pressure correction equation in CFD).

SNUTT Seoul National University Towing Tank

ADI & Other Splitting Methods – Cont.

- Basis of the method: *additive decomposition* of the matrix

$$A = H + V \, , \tag{5.51}$$

  where ***H***: horizontal, i.e., terms contributed by 2nd derivative w.r.t. x and ***V***
- Consider additive **LU** decomposition – different from the multiplicative **LU** decomposition

$$A = L + U \, . \tag{5.52}$$

- Eqs. (5.49) & (5.50)

$$\begin{aligned} (I - L\,\Delta t)\phi^* &= (I + U\,\Delta t)\phi^n \, , \\ (I - U\,\Delta t)\phi^{n+1} &= (I + L\,\Delta t)\phi^* \, . \end{aligned} \tag{5.53}$$

- Each of these steps is essentially a GS iteration.
- Important advantage: may be applied to problems on unstructured grids as well

SNUTT Seoul National University Towing Tank

Iterative Methods – Conjugate Gradient Methods

- Non-linear solvers
  - Newton-like methods: converge quickly *if* an accurate estimate of the solution is available
  - Global methods: guarantee to find the solution, but not very fast
  - Combination: global initially then Newton-like
- Many global methods are descent methods – begin by converting the original eqn system into a minimization problem.

SNUTT Seoul National University Towing Tank

Conjugate Gradient Methods – Cont.

- For positive definite (i.e., symmetric with positive eigenvalues, but not the usual case in CFD) matrices, solving the eqn system (5.1) is equivalent to finding the minimum of

Set the derivative of F or $(\text{eqn})^2$ to zero.

$$F = \frac{1}{2}\phi^T A\phi - \phi^T Q = \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} A_{ij}\phi_i\phi_j - \sum_{i=1}^{n}\phi_i Q_i \qquad (5.54)$$

- Steepest descents (F considered a surface)
  - Oldest & best known method
  - Guaranteed to converge, but often very slowly

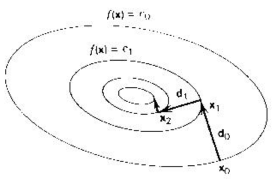


Figure 2.8 Illustration of steepest descent method.

- Conjugate gradient methods
  - New search direction is as different from the old ones as possible

SNUTT Seoul National University Towing Tank

Conjugate Gradient Methods – Cont.

- In 2D, find values of α_1 and α_2 in

  $$\boldsymbol{\phi} = \boldsymbol{\phi}^0 + \alpha_1 \boldsymbol{p}^1 + \alpha_2 \boldsymbol{p}^2 \qquad (5.55)$$

  which minimize F, i.e., we try to minimize F in the $\boldsymbol{p}^1$-$\boldsymbol{p}^2$ plane.
  - In other words, minimize w.r.t. $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ individually provided that the two directions are *conjugate*

    $$\boldsymbol{p}^1 \cdot A\boldsymbol{p}^2 = 0\,. \qquad (5.56)$$
  - The vectors $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$ are said to be conjugate w.r.t. matrix $\boldsymbol{A}$.
- Pre-conditioning – improve the conjugate gradient method
  - Same solution with a smaller condition number $\kappa = \dfrac{\lambda_{\max}}{\lambda_{\min}}$

SNUTT
Seoul National University
Towing Tank

Conjugate Gradient Methods – Cont.

- Pre-multiply the eqn by another matrix

  $$C^{-1}AC^{-1}C\phi = C^{-1}\boldsymbol{Q}\,. \qquad (5.58)$$
- The conjugate gradient method is applied to the modified problem (5.58).
- Algorithm

```
• Initialize by setting: $k = 0$, $\boldsymbol{\phi}^0 = \boldsymbol{\phi}_{\text{in}}$, $\boldsymbol{\rho}^0 = \boldsymbol{Q} - A\boldsymbol{\phi}_{\text{in}}$, $\boldsymbol{p}^0 = \boldsymbol{0}$, $s_0 = 10^{30}$
• Advance the counter: $k = k + 1$
• Solve the system: $M\boldsymbol{z}^k = \boldsymbol{\rho}^{k-1}$
• Calculate: $s^k = \boldsymbol{\rho}^{k-1} \cdot \boldsymbol{z}^k$
      $\beta^k = s^k / s^{k-1}$
      $\boldsymbol{p}^k = \boldsymbol{z}^k + \beta^k \boldsymbol{p}^{k-1}$
      $\alpha^k = s_k / (\boldsymbol{p}^k \cdot A\boldsymbol{p}^k)$
      $\boldsymbol{\phi}^k = \boldsymbol{\phi}^{k-1} + \alpha^k \boldsymbol{p}^k$
      $\boldsymbol{\rho}^k = \boldsymbol{\rho}^{k-1} - \alpha^k A\boldsymbol{p}^k$
• Repeat until convergence.
```

Auxiliary vector

Parameters

Seoul National University
Towing Tank

Conjugate Gradient Methods – Cont.

- This algorithm involves solving a system of linear eqns at the 1st time step. $M = C^{-1}$ where $\boldsymbol{C}$ is the pre-conditioning matrix.
- If $\boldsymbol{M} = \boldsymbol{LU}$ where $\boldsymbol{L}$ and $\boldsymbol{U}$ are the factors used in Stone's SIP method, faster convergence is obtained.

Iterative Methods – Biconjugate Gradients and CGSTAB

- The conjugate gradient method is applicable only to symmetric systems.
  - To apply the method to eqn systems that are not symmetric, convert an asymmetric problem to a symmetric one.
  - Simplest way:

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \cdot \begin{pmatrix} \psi \\ \phi \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q} \\ \boldsymbol{0} \end{pmatrix} . \qquad (5.59)$$

  - When the pre-conditioned conjugate gradient method is applied → Biconjugate gradients
  - It requires almost twice as much effort per iteration as the standard conjugate gradient method, but converges in about the same number of iterations.
  - Other variants…

Iterative Methods – Multigrid Methods

- Basis
  - Iterative method’s rate of convergence depends on the eigenvalues of the iteration matrix associated with the method.
  - The eigenvalues with largest magnitude (spectral radius) determines how rapidly the solution is reached.
  - The eigenvector associated with this eigenvalue determines the spatial distribution of the iteration error and varies considerably from method to method.
  - The iteration error ϵ^n and residual ρ^n after nth iteration are related by

$$A\epsilon^n = \rho^n \, . \qquad (5.15)$$

SNUTT Seoul National University Towing Tank

Multigrid Methods – Cont.

  - If the error is smooth, the update can be computed on a coarser grid.
  - On a grid twice as coarse as the original one in 2D, the iterations cost ¼ as much.
  - Iterative methods converge much faster on coarser grids.
- Much of the work can be done on a coarser grid – we need to define
  - Relationship between the two grids
  - FD operator on the coarse grid
  - Smoothing (restricting) method for the residual from the fine to the coarse grid
  - Interpolating (prolonging) method for the update or correction from the coarse to the fine grid

SNUTT Seoul National University Towing Tank

Multigrid Methods – Cont.

- Example

$$\frac{d^2\phi}{dx^2} = f(x) \tag{5.60}$$

$$\frac{1}{(\Delta x)^2}\left(\phi_{i-1} - 2\phi_i + \phi_{i+1}\right) = f_i \, . \tag{5.61}$$

  - After *n* iterations, approximate solution ϕ^n

$$\frac{1}{(\Delta x)^2}\left(\phi^n_{i-1} - 2\phi^n_i + \phi^n_{i+1}\right) = f_i - \rho^n_i \, . \tag{5.62}$$

  - Subtracting this from Eq. (5.61)

$$\frac{1}{(\Delta x)^2}\left(\epsilon^n_{i-1} - 2\epsilon^n_i + \epsilon^n_{i+1}\right) = \rho^n_i \, , \tag{5.63}$$

Multigrid Methods – Cont.

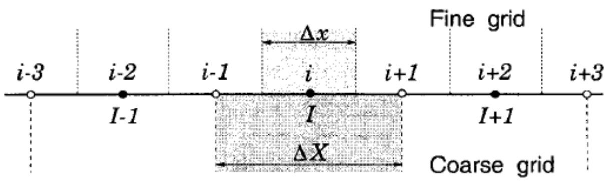


Fig. 5.2. The grids used in the multigrid technique in one dimension

  - Add ½ of Eq. (5.63) with indices *i-1* and *i+1* to the full eqn with index *i*

$$\frac{1}{4\,(\Delta x)^2}\left(\epsilon_{i-2} - 2\epsilon_i + \epsilon_{i+2}\right) = \frac{1}{4}\left(\rho_{i-1} + 2\rho_i + \rho_{i+1}\right) \, . \tag{5.64}$$

  - Using the relationship between the two grids, $\Delta X = 2\,\Delta x$,

$$\frac{1}{(\Delta X)^2}\left(\epsilon_{I-1} - 2\epsilon_I + \epsilon_{I+1}\right) = \overline{\rho}_I \, , \tag{5.65}$$

Multigrid Methods – Cont.

- The simplest prolongation or interpolation of a quantity from the coarse to the fine grid is linear interpolation.
- A 2-grid iterative method

• On the fine grid, perform iterations with a method that gives a smooth error;
• Compute the residual on the fine grid;
• Restrict the residual to the coarse grid;
• Perform iterations of the correction equation on the coarse grid;
• Interpolate the correction to the fine grid;
• Update the solution on the fine grid;
• Repeat the entire procedure until the residual is reduced to the desired level.

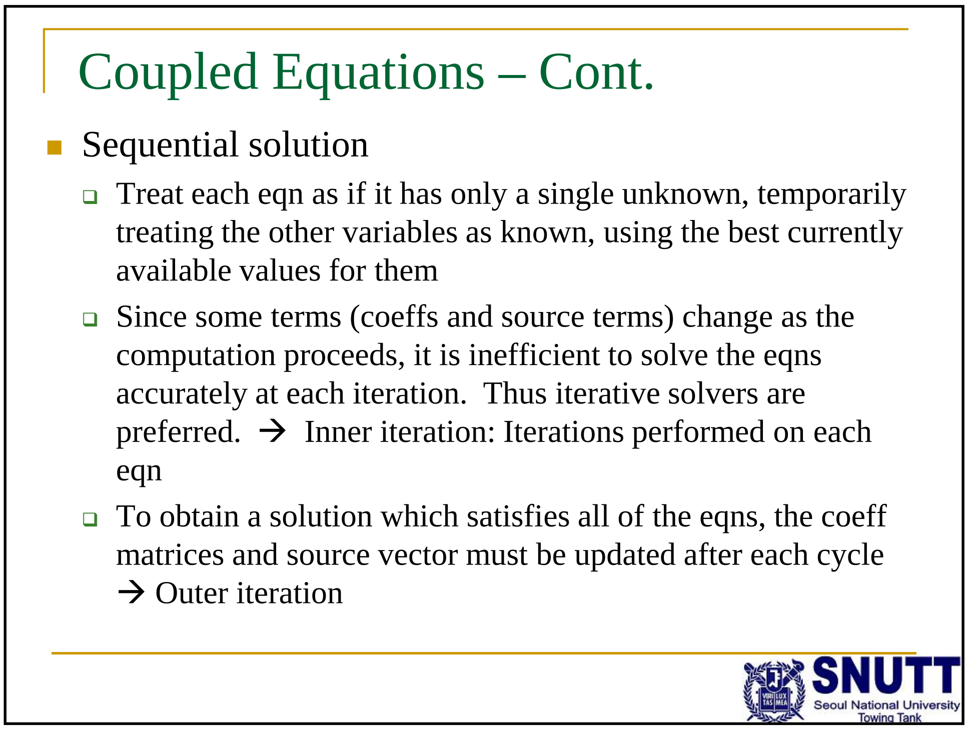
- Full multigrid (FMG) method – for corrections
- Full approximation scheme (FAS) – for solutions

SNUTT Seoul National University Towing Tank

Coupled Equations

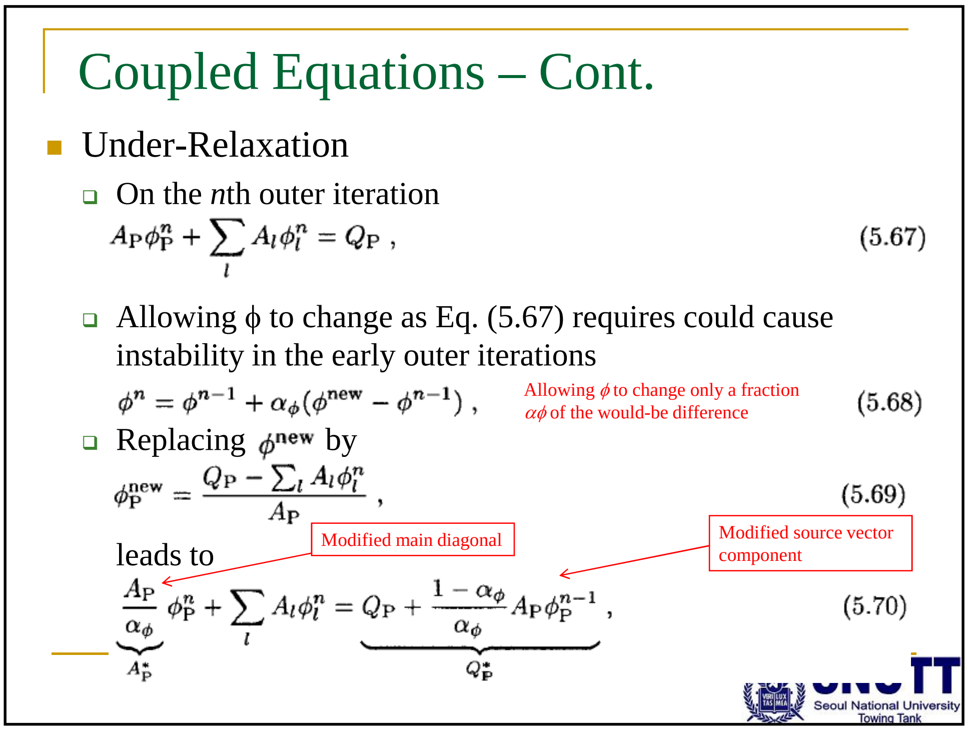
- Most fluid dynamics problems require solution of coupled systems of eqn – dominant variable of each eqn occurs in some of the other eqns
  - All variables are solved for simultaneously
  - Each eqn is solved for its dominant variable, treating the other variables as known
- Simultaneous solution
  - All eqns are considered part of a single system
  - Iterative solution techniques for coupled systems are generalizations of methods for single eqns

SNUTT Seoul National University Towing Tank



Coupled Equations – Cont.

- Sequential solution
  - Treat each eqn as if it has only a single unknown, temporarily treating the other variables as known, using the best currently available values for them
  - Since some terms (coeffs and source terms) change as the computation proceeds, it is inefficient to solve the eqns accurately at each iteration. Thus iterative solvers are preferred. → Inner iteration: Iterations performed on each eqn
  - To obtain a solution which satisfies all of the eqns, the coeff matrices and source vector must be updated after each cycle → Outer iteration



Coupled Equations – Cont.

- Under-Relaxation
  - On the *n*th outer iteration

$$A_P\phi_P^n + \sum_l A_l\phi_l^n = Q_P\ , \tag{5.67}$$

  - Allowing ϕ to change as Eq. (5.67) requires could cause instability in the early outer iterations

$$\phi^n = \phi^{n-1} + \alpha_\phi(\phi^{\text{new}} - \phi^{n-1})\ , \tag{5.68}$$

Allowing ϕ to change only a fraction $\alpha\phi$ of the would-be difference

  - Replacing ϕ^{new} by

$$\phi_P^{\text{new}} = \frac{Q_P - \sum_l A_l\phi_l^n}{A_P}\ , \tag{5.69}$$

leads to

Modified main diagonal

Modified source vector component

$$\underbrace{\frac{A_P}{\alpha_\phi}}_{A_P^*}\phi_P^n + \sum_l A_l\phi_l^n = \underbrace{Q_P + \frac{1-\alpha_\phi}{\alpha_\phi}A_P\phi_P^{n-1}}_{Q_P^*}\ , \tag{5.70}$$

Coupled Equations – Cont.

- Under-Relaxation – Cont.
  - Positive effect, since the diagonal dominance of **A** is increased, i.e., A_P^* is larger than $\boldsymbol{A_P}$
  - More efficient than explicit application of Eq. (5.68)
  - Optimum under-relaxation – problem dependent
  - Start with small and increase towards unity as convergence is approached
  - Under-relaxation may be applied not only to dependent variables, but also to individual terms when the fluid properties depend on the solution and need be updated

Non-Linear Equations and Solutions

- 2 types
  - Newton-like: much faster when a good estimate if available
  - Global: guaranteed not to diverge
  - Trade-off between speed and security
- Newton-like techniques
  - Linearize the function about an estimated value of *x* using

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)\ . \qquad (5.71)$$

  - Setting *f(x)* equal to zero provides a new estimate

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad \text{or, in general,} \quad x_k = x_{k-1} - \frac{f(x_{k-1})}{f'(x_{k-1})} \qquad (5.72)$$

  - Continue until the change in $x_k - x_{k-1}$ is small
  - Equivalent to approximating the *f(x)* curve by its tangent at x_k

Non-Linear Equations – Cont.

- Newton-like techniques – Cont.

$$f_i(x_1, x_2, \ldots, x_n) = 0 , \quad i = 1, 2, \ldots, n . \qquad (5.73)$$

  - Use multi-variable Taylor series

$$f_i(x_1, x_2, \ldots, x_n) = f_i(x_1^k, x_2^k, \ldots, x_n^k) + \sum_{j=1}^{n} (x_j^{k+1} - x_j^k) \frac{\partial f_i(x_1^k, x_2^k, \ldots, x_n^k)}{\partial x_j} , \qquad (5.74)$$

    When this is set to zero → set of linear algebraic eqns

  - Matrix of the system → Jacobian of the system: set of partial derivatives

$$a_{ij} = \frac{\partial f_i(x_1^k, x_2^k, \ldots, x_n^k)}{\partial x_j} , \quad i = 1, 2, \ldots, n , \quad j = 1, 2, \ldots, n , \qquad (5.75)$$

SNUTT Seoul National University Towing Tank

Non-Linear Equations – Cont.

- Newton-like techniques – Cont.
  - The system of eqns is

$$\sum_{j=1}^{n} a_{ij}(x_j^{k+1} - x_j^k) = -f_i(x_1^k, x_2^k, \ldots, x_n^k) , \quad i = 1, 2, \ldots, n . \qquad (5.76)$$

  - To be effective, the Jacobian has to be evaluated at each iteration ← 2 difficulties
    - There are n^2 elements of the Jacobian and their evaluation becomes the most expensive part of the method.
    - Direct evaluating method for the Jacobian may not exist.
    - The cost of generating the Jacobian and solving the system by Gauss elimination is so high that the overall cost is greater than that of other iterative methods.

SNUTT Seoul National University Towing Tank

Non-Linear Equations – Cont.

- Other techniques
  - For the sequential decoupled method, the non-linear terms are usually linearized using *Picard iteration* approach.
  - Non-linear convective term for the u_i momentum component

    $$\rho u_j u_i \approx (\rho u_j)^o u_i \,, \qquad (5.77)$$

  - Source term is decomposed into 2 parts

    $$q_\phi = b_0 + b_1 \phi \,. \qquad (5.78)$$

    - b_0 is absorbed into RHS, while b_1 contributes to **A**.

Deferred Correction Approaches

- Keep the computational molecule as small as possible → storage requirements and linear eqn solution effort
  - Usually nearest neighbors of **P** are kept, but not accurate enough
- Approach 1
  - Leave only the terms containing nearest neighbors on LHS and bring all other to RHS (evaluated using values from previous iteration) → Strong under-relaxation is required to prevent divergence
  - Slow convergence

Deferred Correction Approaches – Cont.

- Approach 2
  - Compute the terms approximated with a high-order approximation explicitly, and put them on RHS
  - Take simpler approximation to these terms, and put it on both LHS (with unknown variable values) and RHS (computing it explicitly using existing values)
  - RHS is now the difference between two, and once converged, the low order approximations terms drop out
  - Used when treating higher-order approximations, grid non-orthogonality, and corrections needed to avoid undesired effects (oscillations).

Deferred Correction Approaches – Cont.

- Approach 2 – Cont.
  - Pade scheme in FD

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1}-\phi_{i-1}}{2\,\Delta x} + \boxed{\left[\left(\frac{\partial \phi}{\partial x}\right)_i^{\text{Padé}} - \frac{\phi_{i+1}-\phi_{i-1}}{2\,\Delta x}\right]^{\text{old}}} . \qquad (5.81)$$

corrections

  - Higher-order flux approximations in FV

$$F_e = F_e^{L} + \boxed{\left(F_e^{H} - F_e^{L}\right)^{\text{old}}} . \qquad (5.82)$$

- Although deferred correction increases the computation time per iteration relative to that for a pure low-order scheme, the additional effort is much smaller than that needed to treat the entire higher-order approximation implicitly.

Convergence Criteria and Iteration Errors

- Important to know when to quit
  - Common procedure: based on the difference between 2 successive iterates, stop when this difference is less than a pre-selected value
  - However the difference may be small when the error is not small → proper normalization is essential
  - From Eqs. (5.14) and (5.27)

$$\boldsymbol{\epsilon}^n = \boldsymbol{\phi} - \boldsymbol{\phi}^n \ , \qquad (5.14)$$

$$\boldsymbol{\epsilon}^n \sim a_1(\lambda_1)^n \boldsymbol{\psi}^1 \ . \qquad (5.27)$$

$$\boldsymbol{\delta}^n = \boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n \approx (\lambda_1 - 1)(\lambda_1)^n a_1 \boldsymbol{\psi}_1 \ , \qquad (5.83)$$

  - The largest eigenvalue or spectral radius, λ_1 , can be estimated

$$\lambda_1 \approx \frac{\|\boldsymbol{\delta}^n\|}{\|\boldsymbol{\delta}^{n-1}\|} \ , \qquad (5.84)$$

SNUTT Seoul National University Towing Tank

Convergence Criteria and Iteration Errors

  - By rearranging Eq. (5.83), iteration error is estimated

$\varepsilon^n \sim a_1(\lambda_1)^n \psi^1$

$$\boldsymbol{\epsilon}^n = \boldsymbol{\phi} - \boldsymbol{\phi}^n \approx \frac{\boldsymbol{\delta}^n}{\lambda_1 - 1} \ . \qquad (5.85)$$

$$\|\boldsymbol{\epsilon}^n\| \approx \frac{\|\boldsymbol{\delta}^n\|}{\lambda_1 - 1} \qquad (5.86)$$

  - This error estimate can be computed from the 2 successive iterates of the solution <- can be quite complex
  - A compromise – Use the reduction of the residual as a stopping criterion → Iteration is stopped when the residual norm has been reduced to some fraction of its original size.
  - The iteration error is related to the residual via Eq. (5.15)

$$A\boldsymbol{\epsilon}^n = \boldsymbol{\rho}^n \ . \qquad (5.15)$$

SNUTT Seoul National University Towing Tank

Convergence Criteria and Iteration Errors

- Experience shows that inner iterations can be stopped when the residual has fallen by 1 - 2 orders of magnitude
- Outer iterations should not be stopped before the residual has been reduced by 3 – 5 orders of magnitude.
- The convergence criterion should be more stringent on refined grids, because the discretization errors are smaller on them than on coarse grids.

SNUTT
Seoul National University
Towing Tank

Examples

- Read through!

SNUTT
Seoul National University
Towing Tank