

# Kapitel 4

## Lösungsverfahren

### 4.1 Iteratives Verfahren der Defektkorrektur

Die Wahl eines Weges zur Lösung eines algebraischen nichtlinearen Gleichungssystems (3.39) bei jedem Zeitschritt  $n + m$  hat einen starken Einfluß auf die Leistungsfähigkeit des verwendeten impliziten Verfahrens für instationäre Strömungen, so daß dafür hocheffiziente Lösungsverfahren entwickelt werden müssen, welche in der Implementierung einen geringen Hauptspeicherbedarf erfordern. In der Literatur existiert eine große Anzahl von Möglichkeiten, derartige Gleichungssysteme zu lösen.

Eine Möglichkeit besteht darin, einen Typ nichtlinearer Relaxationsverfahren in kollektiver Punkt- oder Linienweise anzuwenden, welcher mit dem Verfahren für lineare Gleichungssysteme übereinstimmt. Eine Vereinfachung bei der Anwendung dieser Verfahren ergibt sich dadurch, daß man nicht mehr die Inverse der gesamten Funktionalmatrix benötigt, sondern entsprechend dem jeweiligen Verfahren die Inverse der lokalen Funktionalmatrix. Insbesondere bei großen Systemen ist der geringe Speicherplatzbedarf ein entscheidender Vorteil. Nachteilig ist, daß sie für sehr große Systeme noch nicht schnell genug konvergieren.

Oft werden das *Newton-Raphson*-Verfahren oder dessen Varianten [6], [31], [71] angewendet. Diese Verfahren konvergieren jedoch nur dann mit einer quadratischen Konvergenzrate, wenn die Anfangslösung eine hinreichend genaue Näherung an den Lösungsvektor  $\mathbf{Q}^{n+m}$  ist, und ein lineares Gleichungssystem bei jedem Iterationsschritt direkt oder indirekt exakt gelöst wird. Es ist jedoch in den meisten Fällen schwierig, wegen der komplizierten Strömungsmechanismen die exakte Funktionalmatrix der Vektorfunktion  $\mathbf{Res}(\mathbf{Q}^{n+m})$  zu konstruieren. Da  $\mathbf{Res}(\mathbf{Q}^{n+m})$  von höherer Ordnung ist, besteht außerdem ein großer Speicherplatzbedarf.

Verwendet man alternativ eine Approximation der Vektorfunktion, kann der Lösungsvektor noch wirtschaftlicher gewonnen werden. Aber dafür muß eine geringere Konvergenzrate in Kauf genommen werden. Hier wird das *iterative Verfahren der Defektkorrektur* [3] eingesetzt, welches ein Werkzeug für die Lösung eines nichtlinearen Differenzenoperators höherer Ordnung ist. Bei der Anwendung des Verfahrens auf das Gleichungssystem (3.39) und der Einfachheit halber unter Vernachlässigung des Zeitschrittzählers  $n + m$  ergibt sich

$$\alpha^m V \frac{\mathbf{Q}^{\nu+1}}{\Delta t} - \tilde{\mathbf{N}}(\mathbf{Q}^{\nu+1}) = \mathbf{N}(\mathbf{Q}^\nu) - \tilde{\mathbf{N}}(\mathbf{Q}^\nu) \quad (\nu = 0, 1, 2, \dots), \quad (4.1)$$

wobei  $\tilde{\mathbf{N}}(\mathbf{Q})$  eine Approximation an  $\mathbf{N}(\mathbf{Q})$  darstellt und  $\nu$  den nichtlinearen Iterationszähler. Dieser Iterationsprozeß entspricht der Auflösung des Operators niedriger Ordnung  $\tilde{\mathbf{N}}(\mathbf{Q})$ . Konvergiert dieser Prozeß, dann ergibt er offensichtlich den Lösungsvektor  $\mathbf{Q}^{n+m}$  für den Operator höherer Ordnung  $\mathbf{N}(\mathbf{Q})$ . Die Konvergenzeigenschaft des Prozesses hängt von der Qualität der Approximation  $\tilde{\mathbf{N}}(\mathbf{Q})$  ab.

Gleichung (4.1) selbst ist auch ein nichtlineares System für  $\mathbf{Q}^{\nu+1}$  mit dem Defekt auf der rechten Seite. Daher läßt sich dieses System auch durch ein nichtlineares Relaxationsverfahren oder das *Newton-Raphson*-Verfahren lösen. Hier wird das nichtlineare System (4.1) jedoch in einer linearisierten Form gelöst. Linearisiert man das Gleichungssystem direkt um  $\mathbf{Q}^\nu$  mittels der Taylorreihenentwicklung erster Ordnung, so erhält man ein lineares Gleichungssystem für den Korrekturvektor  $\Delta\mathbf{Q}^\nu (= \mathbf{Q}^{\nu+1} - \mathbf{Q}^\nu)$  der Form

$$\left[ \alpha^m V \frac{I}{\Delta t} + \beta^m \frac{\partial \tilde{\mathbf{Res}}(\mathbf{Q}^\nu)}{\partial \mathbf{Q}^\nu} \right] \Delta\mathbf{Q}^\nu = \mathbf{N}(\mathbf{Q}^\nu) - \frac{V\alpha^m \mathbf{Q}^\nu}{\Delta t} + O[(\Delta\mathbf{Q}^\nu)^2], \quad (4.2)$$

welches für jeden Iterationsschritt  $\nu$  zu lösen ist. Hierbei wird die Funktionalmatrix als nicht-singulär vorausgesetzt. Für  $\tilde{\mathbf{Res}}(\mathbf{Q}^\nu) = \mathbf{Res}(\mathbf{Q}^\nu)$  entspricht dieses Verfahren dem *Newton-Raphson*-Verfahren für (3.39). Die Genauigkeit des Lösungsvektors  $\mathbf{Q}^{n+m}$  wird unabhängig von der Gestaltung der linken Seite allein durch den Defekt auf der rechten Seite bestimmt, und der Linearisierungsfehler  $O[(\Delta\mathbf{Q}^\nu)^2]$  verschwindet, wenn dieser Prozeß konvergiert. Dieses Verfahren ist vollständig konservativ auf der Zeitachse  $t$ , solange es bis zur Konvergenz bei jedem Zeitschritt angetrieben wird. Diese Eigenschaft ist notwendig für instationäre Strömungen, bei welchen die zeitlich genauen Lösungen von Interesse sind.

Für ein Integrationsgebiet aus verschiedenen Typen von Rändern sind geeignete Randbedingungen erforderlich, um das Gleichungssystem (3.39) zu lösen. Alle hier verwendeten Randbedingungen nehmen die Form

$$a(\mathbf{Q}) = a_{soll} \quad (4.3)$$

an, wobei  $a_{soll}$  eine Strömungsgröße oder deren Extrapolationsfunktion auf Rändern eines Integrationsgebiets ist. Ein besseres nichtlineares Konvergenzverhalten kann durch eine implizite Aufprägung von Randbedingungen in die linearisierte Form (4.2) erzielt werden. Um Gleichung (4.3) mit Gleichung (4.2) in Einklang zu bringen, wird Gleichung (4.3) wie folgt linearisiert:

$$\left[ \frac{\partial a(\mathbf{Q}^\nu)}{\partial \mathbf{Q}^\nu} \right] \Delta\mathbf{Q}^\nu = a_{soll} - a(\mathbf{Q}^\nu). \quad (4.4)$$

Im Fall der nichtlinearen Konvergenz ist  $a(\mathbf{Q}) = a_{soll}$ .

Kombiniert man Gleichungen (4.2) und (4.4), so erhält man das lineare Gleichungssystem

$$[A]^\nu \Delta\mathbf{Q}^\nu = \mathbf{RHS}(\mathbf{Q}^\nu) \quad (4.5)$$

mit

$$\mathbf{Q}^{\nu+1} = \mathbf{Q}^\nu + \zeta \Delta\mathbf{Q}^\nu, \quad (4.6)$$

wobei  $\zeta \in (0, 1]$  ein Dämpfungsfaktor zur Stabilisierung des nichtlinearen Konvergenzverhaltens ist. Dieser Koeffizient ist nötig, um die nichtlineare Instabilität zu vermeiden, falls  $\tilde{\mathbf{Res}}(\mathbf{Q})$  mit  $\mathbf{Res}(\mathbf{Q})$  nicht übereinstimmt.  $\mathbf{RHS}(\mathbf{Q}^\nu)$  bringt ein Residuum zum Ausdruck, welches bei jedem Zeitschritt dann verschwindet, wenn der iterative Prozeß (4.5) und (4.6) konvergiert.  $[A]^\nu$  ist die Koeffizientenmatrix, welche schwach besetzt, blockstrukturiert sowie unsymmetrisch ist.

Ein wesentlicher Teil des hohen Berechnungsaufwandes bei impliziten Verfahren ist darauf zurückzuführen, eine Näherung  $\mathbf{Q}^\nu$  an den Lösungsvektor  $\mathbf{Q}^{n+m}$  mit Hilfe der iterativen Vorgehensweise (4.5) und (4.6) bei jedem Zeitschritt aufzufinden, welche  $\mathbf{RHS}(\mathbf{Q}^{n+m}) \approx 0$  ( $\mathbf{Q}^{n+m} \approx \mathbf{Q}^\nu$ ) erfüllt. Im übrigen kostet die Abspeicherung der Koeffizientenmatrix  $[A]^\nu$  einen großen Beitrag zum gesamten Speicherplatz impliziter Verfahren. Ferner sind der Speicherplatzbedarf und das Konvergenzverhalten von Lösungsverfahren für (4.5) sehr unterschiedlich. Unter Berücksichtigung der Leistungsfähigkeit und des Hauptspeicherplatzes des zur Verfügung stehenden Rechners muß ein optimales Lösungsverfahren gewählt werden.

Eine gute Anfangsbelegung der Strömungsgrößen bei jedem Zeitschritt reduziert die Anzahl der erforderlichen nichtlinearen Iterationen bis zur Konvergenz, so daß die Rechenkosten von der Qualität der Anfangsnäherung  $\mathbf{Q}^{\nu=0}$  beeinflußt werden. Die Anfangsnäherung wird hier durch die Beziehung

$$\mathbf{Q}^{\nu=0} = \sum_{j=0}^m \beta^j \mathbf{Q}^{n+j-1} + \sum_{j=0}^m \alpha^j \mathbf{Q}^{n+j-1} \quad (4.7)$$

bei jedem Zeitschritt abgeschätzt, welche eine ähnliche Form wie (3.39) annimmt.

Zur Beurteilung einer konvergierten Näherung  $\mathbf{Q}^\nu$  an den Lösungsvektor  $\mathbf{Q}^{n+m}$  muß ein Kriterium herangezogen werden. Das hier verwendete Kriterium basiert auf dem *Root-Mean-Square (RMS)*:

$$\left( \frac{1}{N} \sum_{i=0}^N \mathbf{RHS}^2(\mathbf{Q}^\nu) \right)^{1/2} \leq \phi, \quad (4.8)$$

wobei  $N$  die Anzahl der Gitterpunkte in einem Integrationsgebiet ist. Der nächste Zeitschritt wird dann gestartet, wenn ein Wert des *RMS* eine voreingestellte Schranke  $\phi$  unterschreitet. Die Rechenkosten sowie die zeitliche Lösungsgenauigkeit sind von der Wahl der Schranke stark abhängig.

Falls nur stationäre Strömungen von Interesse sind, kann man ein nichtiteratives Verfahren erster Ordnung noch effektiver nutzbar machen. Für  $m = 1$ ,  $\alpha^1 = -\alpha^0 = 1$ ,  $\beta^1 = 1$ ,  $\beta^0 = 0$  erhält man aus (4.2)

$$\left[ V \frac{I}{\Delta t} + \frac{\partial \tilde{\mathbf{Res}}(\mathbf{Q}^\nu)}{\partial \mathbf{Q}^\nu} \right] \Delta \mathbf{Q}^\nu = \frac{V(\mathbf{Q}^\nu - \mathbf{Q}^n)}{\Delta t} - \mathbf{Res}(\mathbf{Q}^n) \quad (4.9)$$

und für  $\mathbf{Q}^{\nu=0} = \mathbf{Q}^n$  erhält man schließlich das *Euler-rückwärts*-Verfahren in der nichtiterativen Form

$$\left[ V \frac{I}{\Delta t} + \frac{\partial \tilde{\mathbf{Res}}(\mathbf{Q}^n)}{\partial \mathbf{Q}^n} \right] \Delta \mathbf{Q}^n = -\mathbf{Res}(\mathbf{Q}^n), \quad (4.10)$$

welches für die Berechnung stationärer Strömungen heutzutage häufig zum Einsatz kommt. Dieses nichtiterative *Euler-rückwärts*-Verfahren ist bezüglich der Zeitachse nicht konservativ und ergibt die Lösungen, welche die Erhaltungseigenschaft nur im stationären Zustand erfüllen. Für  $\Delta t \rightarrow \infty$  entspricht dieses Verfahren dem iterativen Verfahren der Defektkorrektur in linearisierter Form für die diskreten stationären Erhaltungsgleichungen  $\mathbf{Res}(\mathbf{Q})$ . Als Folge kann man das Verfahren als ein Lösungsverfahren mit einem Parameter  $I/\Delta t$  bezeichnen, welchen man einführt, um die nichtlineare Konvergenz sicherzustellen. Für eine Erhöhung der Konvergenzrate kann man eine bei jedem Zeitschritt lokal veränderliche Zeitschrittgröße  $\Delta t$  aus der *CFL*-Stabilitätsbedingung einsetzen.

Das nichtiterative *Euler-rückwärts*-Verfahren (4.10) wird zusammen mit der Randbedingung (4.4) wie folgt formuliert:

$$[A]^n \Delta \mathbf{Q}^n = \mathbf{RHS}(\mathbf{Q}^n) \quad (4.11)$$

mit

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta \mathbf{Q}^n, \quad (4.12)$$

wobei  $\mathbf{RHS}(\mathbf{Q})$  ein Residuum ist, welches im stationären Zustand verschwindet und damit die räumliche Genauigkeit der stationären Strömungsgrößen bestimmt, also die auskonvergierte Lösung ist völlig unabhängig von der Zeitachse.

Die Lösung linearer Gleichungssysteme gehört zu einer zeitaufwendigen Aufgabe in impliziten Verfahren. Lösungsverfahren für das Gleichungssystem (4.5) um den Korrekturvektor  $\Delta \mathbf{Q}'$  lassen sich grob in direkte und iterative Wege unterteilen. Allerdings koppelt man auch oft ein direktes Verfahren mit einem iterativen. Welches Verfahren zur Verwendung kommt, kommt auf die Umstände an, da jedes Verfahren in verschiedenen Punkten eigene Vor- und Nachteile bringt.

Direkte Verfahren wie z. B. der *Gaußsche* Algorithmus oder das *Cholesky*-Verfahren [71] liefern eine gute Näherung  $\Delta \hat{\mathbf{Q}}'$  an  $\Delta \mathbf{Q}'$  in endlich vielen Rechenschritten, allerdings nur dann, wenn man unrealistisch rundungsfreie Rechnung voraussetzt. Rundungsfehler wirken sich bei direkten Verfahren immer in Form einer Verfälschung der Lösung aus. Hierdurch kann die numerisch bestimmte Lösung erheblich verfälscht sein, was sich in einigen Fällen durch die sogenannte *Nachiteration* beheben oder mindern läßt. Darüber hinaus sind direkte Verfahren für schwach besetzte Matrizen nicht geeignet, da im Verlauf der Rechenschritte neue Matrizenelemente entstehen, d. h. nicht besetzte Elemente werden aufgefüllt. Dies ist bei iterativen Wegen nicht der Fall. Außerdem sind direkte Verfahren mit Sicherheit keine leistungsfähigen Lösungsverfahren für das Gleichungssystem (4.5), denn die nichtlineare Konvergenz des *Newton-Raphson*-Verfahrens kann wegen der unterschiedlichen Operatoren auf der linken und rechten Seiten nicht erreicht werden. Das Ziel der Lösung für das Gleichungssystem (4.5), besteht hier darin, eine gute Näherung an  $\Delta \mathbf{Q}'$  nicht zu errechnen, sondern mit Hilfe einer geeigneten Näherung an  $\Delta \mathbf{Q}'$  so schnell wie möglich eine Näherung  $\mathbf{Q}'$  herauszufinden, welche  $\mathbf{RHS}(\mathbf{Q}') \approx 0$  erfüllt. Es wird daher an die Lösungsgenauigkeit keine zu hohen Forderungen gestellt. Daneben nehmen direkte Verfahren noch viel Speicherplatz ein.

Eine Alternative dazu ist das *angenäherte Faktorisierungsverfahren* nach *Beam* und *Warming* [8], bei welchem man den impliziten Operator, d. h. die Funktionalmatrix in (4.2) zu den eindimensionalen Operatoren in jeder Koordinatenrichtung annähernd so faktorisiert,

daß jedes einzelne Gleichungssystem leicht direkt gelöst werden kann. Ein Nachteil dieses Verfahrens ist, daß eine ungenaue Näherung an  $\Delta \mathbf{Q}'$  infolge des Faktorisierungsfehlers unter Umständen ein Konvergenzproblem mit sich bringen kann. Andererseits ist das Verfahren vollständig vektorisierbar, und dessen Anwendbarkeit ist unabhängig von der Art der Flußberechnung.

So kommen auch iterative Wege in Betracht, bei welchen ausgehend von einer Anfangsnäherung  $\Delta \tilde{\mathbf{Q}}'$  an  $\Delta \mathbf{Q}'$  die Näherung iterativ verbessert wird. Iterative Verfahren sind dann geeignet, wenn die Koeffizientenmatrix eines Gleichungssystems schwach besetzt ist und gewisse spezielle Eigenschaften besitzt, welche die Konvergenz gewährleisten. Dies trifft insbesondere bei dem Gleichungssystem (4.5) zu. Für eine Näherung  $\Delta \tilde{\mathbf{Q}}'$ , durch welche der Korrekturvektor  $\Delta \mathbf{Q}'$  in (4.5) ersetzt wird, sollte das Gleichungssystem deshalb vorzugsweise iterativ gelöst werden.

Die Anwendbarkeit dieser Technik geht davon aus, daß ein *approximativer Riemannlöser* erster Ordnung für die reibungsfreie Flußfunktion eine gut konditionierte Koeffizientenmatrix  $[A]'$  für iterative Vorgehensweisen liefert. Die Iterationstechnik ist nicht anwendbar auf die Gleichungssysteme mit Koeffizientenmatrizen, welche sich aus dem zentralen Verfahren für die Flußberechnung ergeben, da die Koeffizientenmatrizen für Iterationsverfahren schlecht konditioniert sind, so daß in diesem Fall auf ein direktes Verfahren oder das *angenäherte Faktorisierungsverfahren* zurückgegriffen werden muß. Für die Iterationstechnik spricht in erster Linie der minimale Speicherplatzbedarf, zusätzlich ist sie in den meisten Fällen für einen großen Zeitschritt noch stabiler. Diese Iterationstechnik ist aber mit mehreren Ausnahmen (z. B. die *Jacobi*-Typ- oder die *Gauss-Seidel*-Typ-Relaxation mit Schachbrettanordnung) nicht vollständig vektorisierbar.

Die Leistungsfähigkeit eines Lösungsverfahrens kann nach den Kriterien der benötigten Rechenzeit sowie des Speicherplatzbedarfs beurteilt werden. Welches Verfahren jedoch Anwendung findet, hängt von vielen Faktoren ab, darunter von den Rechnern, von der Eigenschaft der Funktionalmatrix und von der Genauigkeit, welche vom Korrekturvektor  $\Delta \mathbf{Q}'$  verlangt wird. In dieser Arbeit wurde für die Gestaltung der Funktionalmatrix in (4.2) nur die reibungsfreie numerische Flußfunktion  $\tilde{\mathbf{F}}_I^{\xi_i}$  in (3.30) verwendet, welche anhand des *approximativen Riemannlösers* erster Ordnung nach *Roe* [61] erzielt wurde, so daß im folgenden die von der Iterationstechnik herrührenden, verschiedenen Lösungsverfahren für das Gleichungssystem(4.5) erläutert werden sollen.

### 4.1.1 Klassische Iterationsverfahren

Werden zuerst zeilen- bzw. spaltenweise sämtliche Gitterpunkte in einem Rechengitter durchnummeriert, so sammeln sich die Elemente der Koeffizientenmatrix  $[A]'$  in Blöcken. Die Koeffizientenmatrix  $[A]'$  nimmt eine schwach besetzte  $N \times N$  Blockmatrix an, da Verknüpfungen zwischen den Zustandsgrößen an einem Knotenpunkt nur mit den jeweils benachbarten Knotenpunkten vorkommen. Die Dimension jedes Blocks läßt sich von der Anzahl der Erhaltungsgrößen bestimmen. Die Elemente innerhalb der Blöcke selbst können unter Umständen mit Ausnahme der diagonalen Elemente Null sein. Nur die Nichtnull-Elemente innerhalb der Blöcke werden durch eine geschickte Speichertechnik für schwach besetzte Matrizen zeilenwei-

se abgespeichert.

Wie schon oben erwähnt, geht der Wunsch zum Einsatz von Iterationsverfahren für (4.5) daraus hervor, daß ein *approximativer Riemannlöser* erster Ordnung für die reibungsfreie Flußfunktion eine beinahe blockdiagonaldominante Koeffizientenmatrix  $[A]^\nu$  liefert. Diese hervorragende Bedingung der Koeffizientenmatrix erlaubt, Iterationsverfahren zur Lösung des Gleichungssystems (4.5) heranzuziehen. Es ist jedoch bekannt, daß im Fall höherer Ordnung diese Bedingung mit zunehmender Zeitschrittgröße  $\Delta t$  verloren geht [76].

Der Mehrzahl praktisch verwendeter Iterationsverfahren für lineare Gleichungssysteme liegt im Prinzip folgendes zugrunde: Unter der Annahme, daß eine nichtsinguläre Matrix  $[K]^\nu$  existiert, welche leicht invertierbar und eine Approximation an die Koeffizientenmatrix  $[A]^\nu$  ist, kann man das Gleichungssystem (4.5) auf die Form

$$[K]^\nu \Delta \mathbf{Q}^\nu = ([K]^\nu - [A]^\nu) \Delta \mathbf{Q}^\nu + \mathbf{RHS}(\mathbf{Q}^\nu) \quad (4.13)$$

bringen, welche ausgehend von einer Ausgangsnäherung  $\Delta \mathbf{Q}^{\nu, \kappa=0}$  zum iterativen Prozeß der Form

$$[K]^\nu \Delta \mathbf{Q}^{\nu, \kappa+1} = ([K]^\nu - [A]^\nu) \Delta \mathbf{Q}^{\nu, \kappa} + \mathbf{RHS}(\mathbf{Q}^\nu), \quad (4.14)$$

$$[K]^\nu \delta^\kappa = \mathbf{RHS}(\mathbf{Q}^\nu) - [A]^\nu \Delta \mathbf{Q}^{\nu, \kappa} = \mathbf{r}^\kappa \quad (\kappa = 0, 1, 2, \dots) \quad (4.15)$$

mit der Korrektur  $\delta^\kappa$

$$\Delta \mathbf{Q}^{\nu, \kappa+1} = \omega \delta^\kappa + \Delta \mathbf{Q}^{\nu, \kappa} \quad (4.16)$$

sowie dem Defekt  $\mathbf{r}^\kappa$  führt, wobei  $\kappa$  der lineare Iterationszähler ist und  $\omega \in (0, 2]$  ein Parameter zur schnellen Minimierung einer Norm des Defekts  $\mathbf{r}^\kappa$ , d. h. zur Verbesserung der Konvergenzgeschwindigkeit. Man nennt  $\omega$  *Relaxationsparameter*.

Die Anfangsnäherung  $\Delta \mathbf{Q}^{\nu, \kappa=0}$  des Iterationsprozesses (4.15) und (4.16) ist beliebig, jedoch ist

$$\Delta \mathbf{Q}^{\nu, \kappa=0} = \Delta \tilde{\mathbf{Q}}^{\nu-1} \quad (4.17)$$

eine gute Wahl. Die Koeffizientenmatrix  $[K]^\nu$  wird als die *Zerlegungsmatrix* bezeichnet, von deren Wahl ein Iterationsverfahren abhängt. Für  $[K]^\nu = [A]^\nu$  bzw.  $\Delta \mathbf{Q}^{\nu, \kappa=0} = 0$  wird der Iterationsprozeß zu einem direkten Verfahren.

Da bei jedem Zeitschritt  $n+m$  nur die nichtlineare Konvergenz zur Auffindung einer Näherung  $\mathbf{Q}^\nu$  an  $\mathbf{Q}^{n+m}$ , welche  $\mathbf{RHS}(\mathbf{Q}^\nu) \approx 0$  erfüllt, von Interesse ist, ist es unnötig, das Gleichungssystem (4.5) bei jedem Iterationsschritt  $\nu$  mit übermäßiger Genauigkeit aufzulösen. Nur eine Approximation  $\Delta \tilde{\mathbf{Q}}^\nu$  an  $\Delta \mathbf{Q}^\nu$  des Gleichungssystems (4.5) ist gesucht. Auch das Abbruchkriterium des Iterationsprozesses (4.15) hat also einen wesentlichen Einfluß auf die Rechenkosten bei der Bestimmung der Nullstellen der Vektorfunktion  $\mathbf{RHS}(\mathbf{Q}^\nu)$ . Es wird umso mehr Rechenzeit bei jedem Iterationsschritt  $\nu$  erforderlich, je genauer man das Gleichungssystem (4.5) auflöst.

Es stellt sich hier die Frage, in welchem Umfang das Gleichungssystem (4.5) aufgelöst werden soll, um die beste nichtlineare Konvergenz zu erhalten. Hierfür benötigt man ein geeignetes Abbruchkriterium. Wann der Iterationsprozeß abzubrechen ist, kann man entweder

vor Beginn der Rechnung festlegen (*starre* Technik) oder im Verlauf der Rechnung anhand der erzielten Genauigkeit entscheiden (*adaptive* Technik). Bei der starren Technik stellt man im voraus die Anzahl der Iterationen  $\kappa$  ein. Das Gleichungssystem (4.5) wird mit übermäßiger Genauigkeit aufgelöst, wenn  $\kappa$  zu groß ist. Umgekehrt wird es zu ungenau aufgelöst, wenn  $\kappa$  zu klein ist. Es ist schwierig, eine optimale Anzahl  $\kappa$  zu bestimmen, welche zu einer zufriedenstellenden nichtlinearen Konvergenz führt, denn die optimale Anzahl  $\kappa$  ist abhängig von vielen Faktoren. Bei der adaptiven Technik wird der Iterationsprozeß unabhängig von  $\kappa$  abgebrochen, wenn eine Näherung  $\Delta\tilde{\mathbf{Q}}^\nu$  des Gleichungssystems (4.5) ein vorgegebenes Kriterium erfüllt. In dieser Arbeit wird der Iterationsprozeß dann abgebrochen, wenn die Bedingung

$$\frac{\|\mathbf{r}^\kappa\|_2}{\|\mathbf{RHS}(\mathbf{Q}^\nu)\|_2} \leq \phi \quad (4.18)$$

erfüllt ist, wobei  $\phi \in [0, 1)$  einen Faktor darstellt, welcher die Lösungsgenauigkeit des Gleichungssystems (4.5) kontrolliert und  $\|\cdot\|_2$  die 2-Norm eines Vektors zum Ausdruck bringt. Mit abnehmendem Wert von  $\phi$  wird zwar eine genauere Näherung an  $\Delta\mathbf{Q}^\nu$  erzielt, aber es werden immer mehr Iterationen  $\kappa$  erforderlich.  $\phi = 0$  bedeutet das exakte Auflösen des Gleichungssystems (4.5).

Der Faktor  $\phi$  liefert einen Austausch zwischen der Genauigkeit, mit welcher das Gleichungssystem (4.5) aufgelöst wird und der Arbeitsleistung pro Iterationsschritt  $\nu$ . Die Wahl von  $\phi$  hängt von vielen Faktoren ab, wie etwa von der Form des impliziten oder expliziten Operators, der *Courant*-Zahl, den Eigenschaften der in Frage kommenden Strömungen. Hierbei hat sich insgesamt  $\phi = 0.1$  als eine gute Wahl bewährt [25]. Falls eine Näherung  $\Delta\tilde{\mathbf{Q}}^\nu$  das Abbruchkriterium (4.18) erfüllt, bringt der Defekt  $\mathbf{r}^\kappa$  eine Größe zum Ausdruck, um welche die Näherung die Erfüllung von (4.5) verfehlt.

Der iterative Prozeß erfolgt in kollektiver Weise, bei welcher die an jedem Gitterpunkt oder jeder Gitterlinie auftretenden Unbekannten gleichzeitig aktualisiert werden, und individuelle Komponenten durch andere Komponenten in einem Gitterpunkt oder einer Gitterlinie implizit definiert sind. Dieser kollektive iterative Prozeß stellt eine Mischung von direktem und iterativem Verfahren dar und führt oft zu einer Zunahme an Konvergenz auf Kosten gewisser Komplikationen im rechnerischen Algorithmus. Um zu einem Iterationsverfahren zu gelangen, wird zunächst die Koeffizientenmatrix  $[A]^\nu$  wie folgt zerlegt:

$$[A]^\nu = [L]^\nu + [D]^\nu + [U]^\nu, \quad (4.19)$$

wobei  $[D]^\nu$  eine leicht invertierbare Diagonalmatrix von  $[A]^\nu$  ist, und  $[L]^\nu$  und  $[U]^\nu$  jeweils die Dreiecksmatrizen unter- und oberhalb von  $[D]^\nu$  sind. Entsprechend der Wahl von  $[D]^\nu$  unterscheidet man zwischen *Punkt*- und *Linien*-Iterationsverfahren.

Zunächst werden *Punkt*-Iterationsverfahren erläutert, welche auf strukturierte sowie unstrukturierte Gitter anwendbar und mit einer direkten Lösung eines Untergleichungssystems mit Blockmatrix  $[D]^\nu$  an jedem Gitterpunkt verbunden sind. Hierbei wird die Blockmatrix vollständig faktorisiert. Eine wesentliche Rechenzeit wird für die Faktorisation der Blockmatrix an jedem Gitterpunkt aufgewendet. So wird die Blockdiagonalmatrix während des Iterationsprozesses (4.15) nur einmal faktorisiert und abgespeichert, weil sich die Zerlegungsmatrix  $[K]^\nu$  für einen Iterationsschritt  $\nu$  im Gegensatz zum Defekt  $\mathbf{r}^\kappa$  nicht verändert. Die Blockmatrix an jedem Gitterpunkt wird zuerst in eine Links- und eine Rechtsdreiecksmatrix

zerlegt. Es folgt darauf ein Vorwärts- und Rückwärtseinsetzen. Dies verhindert zahlreiche Faktorisierungen während des Iterationsprozesses (4.15). Dies ist ein wichtiger Punkt für die Rechenzeitersparnis.

Eines der einfachsten Iterationsverfahren ist das *Punkt-Jacobi*-Verfahren, bezeichnet auch als *Gesamtschritt-Verfahren*, dessen Zerlegungsmatrix die Form

$$[K]^\nu = [D]^\nu \quad (4.20)$$

annimmt. Beim *Punkt-Jacobi*-Verfahren kann die Korrektur  $\delta^\kappa$  in (4.15) unabhängig voneinander und in beliebiger Reihenfolge berechnet werden. Da keine Rekursionen auftreten, ist das Verfahren sehr gut vektorisierbar.

Das *Punkt-Jacobi*-Verfahren nutzt die aktualisierte Korrektur  $\delta^\kappa$  in einem Iterationsschritt  $\kappa$  nicht aus, so daß dieses Schema etwas langsam konvergiert. Im Gegensatz dazu macht das *Punkt-Gauss-Seidel*-Verfahren von der an den vorhergehenden Knotenpunkten errechneten Korrektur  $\delta^\kappa$  in einem Iterationsschritt  $\kappa$  Gebrauch, um die Konvergenzrate zu verbessern. Aufgrund der Rekursivität des Verfahrens ist es jedoch schlecht zu vektorisieren. Beim *Punkt-Gauss-Seidel*-Verfahren ist die Zerlegungsmatrix durch

$$[K]^\nu = [L]^\nu + [D]^\nu \quad (4.21)$$

gegeben. Man nennt das *Punkt-Gauss-Seidel*-Verfahren deshalb auch *Einzel-schritt-Verfahren*.

Der iterative Prozeß des *symmetrischen Punkt-Gauss-Seidel*-Verfahrens besteht aus den zwei *Gauss-Seidel*-Schritten, d. h. den vorwärtigen und rückwärtigen Schritten [4]:

$$([L]^\nu + [D]^\nu)(\Delta \mathbf{Q}^{\nu, \kappa+1/2} - \Delta \mathbf{Q}^{\nu, \kappa}) = \mathbf{RHS}(\mathbf{Q}^\nu) - [A]^\nu \Delta \mathbf{Q}^{\nu, \kappa}, \quad (4.22)$$

$$([D]^\nu + [U]^\nu)(\Delta \mathbf{Q}^{\nu, \kappa+1} - \Delta \mathbf{Q}^{\nu, \kappa+1/2}) = \mathbf{RHS}(\mathbf{Q}^\nu) - [A]^\nu \Delta \mathbf{Q}^{\nu, \kappa+1/2}. \quad (4.23)$$

Hieraus errechnet man die Zerlegungsmatrix  $[K]^\nu$  der Form

$$[K]^\nu = ([L]^\nu + [D]^\nu)[D^{-1}]^\nu([D]^\nu + [U]^\nu). \quad (4.24)$$

Wie oben schon erwähnt, wird die Blockmatrix an jedem Gitterpunkt  $[D]^\nu$  nur einmal faktorisiert. Die Multiplikation der Blockdiagonalmatrix mit einem Vektor erfolgt dagegen in einer so geschickten Weise, daß die schon faktorisierte Matrix  $[D]^\nu$  unberührt erhalten werden kann. Das *symmetrische Punkt-Gauss-Seidel*-Verfahren erfordert pro Iteration etwa zweimal soviel Arbeitsleistung wie das *Punkt-Gauss-Seidel*-Verfahren. Aber dafür ist das symmetrische Verfahren für einen Vorkonditionierer in einem *konjugierten Gradienten*-Typ-Verfahren effektiv einsetzbar und zeigt in den meisten Fällen noch schnellere Konvergenz als bei Verwendung des *Punkt-Gauss-Seidel*-Verfahrens.

Bei dem Verfahren der *unvollständigen Punkt-Zerlegung* nimmt die Zerlegungsmatrix die Form

$$[K]^\nu = ([L]^\nu + [D_0]^\nu)[D_0^{-1}]^\nu([D_0]^\nu + [U]^\nu) \quad (4.25)$$

an. Diese Matrix läßt sich wie folgt entwickeln:

$$[K]^\nu = [L]^\nu + [D_0]^\nu + [U]^\nu + [L]^\nu[D_0^{-1}]^\nu[U]^\nu. \quad (4.26)$$



Aus (4.19) und (4.26) folgt die Beziehung

$$[A]^\nu - [K]^\nu = [D]^\nu - [D_0]^\nu - [L]^\nu [D_0^{-1}]^\nu [U]^\nu. \quad (4.27)$$

Berechnet man  $[D_0]^\nu$  in (4.27) exakt, so daß  $([A]^\nu - [K]^\nu) = 0$  erfüllt ist, so entspricht (4.25) einer *vollständigen Punkt-Zerlegung*. Die *unvollständige Punkt-Zerlegung* erfordert, daß das Blockdiagonal von  $([A]^\nu - [K]^\nu)$  gleich Null ist:

$$[D_0]^\nu = [D]^\nu - \text{Blockdiagonal}([L]^\nu [D_0^{-1}]^\nu [U]^\nu), \quad (4.28)$$

welche im Gegensatz zum *symmetrischen Punkt-Gauss-Seidel-Verfahren* noch mehr Rechenzeit und Speicherplatz benötigt. Für  $[D_0]^\nu = [D]^\nu$  wird (4.25) zur Zerlegungsmatrix des *symmetrischen Punkt-Gauss-Seidel-Verfahrens*. Auch die Berechnung und Faktorisierung der Blockdiagonalmatrix  $[D_0]^\nu$  erfolgt nur einmal vor Beginn des iterativen Prozesses (4.15). Dieses Verfahren wird daher auch als Vorkonditionierer in *konjugierten Gradienten-Typ-Verfahren* oder als Glätter in Mehrgitterverfahren häufig verwendet.

Numeriert man die Gitterpunkte eines strukturierten Gitternetzes wie auf dem Schachbrett, dann entsteht ein reduziertes System, bei welchem der iterative Prozeß (4.15) auf den geraden und ungeraden Punkten völlig entkoppelt ist. Führt man für ein durchnummeriertes Gitternetz den iterativen Prozeß zuerst auf den ungeraden Punkten und daraufhin auf den geraden Punkten durch, so erhält man ein reduziertes System mit dem gleichen Effekt, wie eine Schachbrettanordnung. Die jeweiligen Zerlegungsmatrizen auf den ungeraden und geraden Punkten nehmen

$$[K_u]^\nu = [D]^\nu, \quad [K_g]^\nu = [L]^\nu + [D]^\nu + [U]^\nu \quad (4.29)$$

an. Im ersten Schritt wird der iterative Prozeß (4.15) in gleicher Weise, wie das *Punkt-Jacobi-Verfahren* implementiert. Im zweiten Schritt ist die Korrektur  $\delta^k$ , entsprechend den Matrizen  $[L]^\nu$  und  $[U]^\nu$ , aus dem ersten Schritt bekannt, so daß die Implementierung von (4.15) nach  $[K_g]^\nu$  ohne Schwierigkeit vorgenommen werden kann. Für ein derartiges System muß jedoch die Anzahl der Gitterpunkte an der Gitterlinie, entlang derer die Gitterpunkte durchnummeriert werden, ungerade sein. Durch diese Schachbrettanordnung kann man auch zu einer guten Vektorisierung kommen.

In manchen Fällen kann die Konvergenz der bisher dargestellten Iterationsverfahren dadurch beschleunigt werden, daß die Iteration linienweise ausgeführt wird. Bei *Linien-Verfahren* werden die Matrizenelemente entlang einer Gitterlinie zu  $[D]^\nu$  zusammengefaßt, welche blocktridiagonal ist. An den Gitterlinien, an denen Überschallströmungen herrschen oder sich Ränder befinden, ist jedoch  $[D]^\nu$  nicht mehr blocktridiagonal. Anders als *Punkt-Verfahren* verbessert man mit *Linien-Verfahren* die Approximation an  $\Delta \mathbf{Q}^\nu$  gleichzeitig entlang jeder Gitterlinie, was also eine direkte Auflösung eines Gleichungssystems mit Koeffizientenmatrix  $[D]^\nu$  entlang jeder Gitterlinie erfordert. Die Matrix  $[D]^\nu$  wird wie im Fall der *Punkt-Verfahren* nur einmal faktorisiert. Die Iterationsvorschriften der *Punkt-Verfahren* sind direkt übertragbar auf *Linien-Verfahren*. Im Vergleich zu *Punkt-Verfahren* können *Linien-Verfahren* auf unstrukturierte Gitter nicht angewendet werden, und erfordern wegen ihrer großen Zerlegungsmatrix noch mehr Speicherplatz.

Für strukturierte Gitter wird die Anordnung eines Gitternetzes schon bei der Gittergenerierung festgelegt. Da jedoch ein iteratives Verfahren eine ausgezeichnete Koordinatenrichtung

besitzt, sollte der iterative Prozeß so flexibel sein, daß er in jeder Koordinatenrichtung vorgenommen werden kann. Ordnet man die entlang einer Koordinatenrichtung abgespeicherte Diagonalmatrix  $[D]^\nu$  in eine andere Koordinatenrichtung um, so können alle bis jetzt erläuterten Lösungsverfahren auf die andere Koordinatenrichtung angewendet werden. Bei *Linien-*Verfahren ergibt sich eine deutliche Rechenzeiterparnis durch die wechselseitige Ausführung der Iteration in jeder Koordinatenrichtung. Bei *alternierenden Linien-*Verfahren muß man in jeder Koordinatenrichtung die Diagonalmatrix  $[D]^\nu$  bei jeder Iteration  $\kappa$  faktorisieren. Reserviert man allerdings die Speicherplätze für  $[D]^\nu$  in jeder Koordinatenrichtung, so brauchen die Verfahren die Diagonalmatrix  $[D]^\nu$  nur einmal zu faktorisieren. Somit muß man sich für Rechenzeit oder Speicherplatz entscheiden.

Aus der Gleichung (4.14) erhält man die Fixpunktform

$$\Delta \mathbf{Q}^{\nu, \kappa+1} = [K^{-1}]^\nu ([K]^\nu - [A]^\nu) \Delta \mathbf{Q}^{\nu, \kappa} + [K^{-1}]^\nu \mathbf{RHS}(\mathbf{Q}^\nu). \quad (4.30)$$

Ein Iterationsverfahren ist für jede Anfangsnäherung höchstens dann noch konvergent, wenn der Spektralradius der Iterationsmatrix  $[K^{-1}]^\nu ([K]^\nu - [A]^\nu)$  kleiner als 1 ist. Dies hängt jedoch bei klassischen Iterationsverfahren noch von der Gitterdichte und damit von der Größe des betrachteten Gleichungssystems ab. Für große Gleichungssysteme liegt der Spektralradius wenig unter 1, d. h. die Konvergenzrate ist entsprechend gering.

### 4.1.2 Konjugierte Gradienten-Typ-Verfahren

In klassischen Iterationsverfahren hängt ihre Konvergenzrate von der Wahl des Parameters  $\omega$  in (4.16) ab. Es ist jedoch in der Praxis schwierig, den optimalen Wert analytisch zu bestimmen, so daß die Wahl meistens durch rechnerische Experimente getroffen wird. Man versucht, eine Folge von  $\omega^\kappa$  derart auszuwählen, daß  $\|\mathbf{r}^\kappa\|_2$  rapide minimiert werden kann. Viele Minimierungsverfahren für die Korrektur  $\delta^\kappa$  in (4.16) können zur Form

$$\Delta \mathbf{Q}^{\nu, \kappa+1} = \omega^\kappa \delta^\kappa + \Delta \mathbf{Q}^{\nu, \kappa} \quad (4.31)$$

zusammengefaßt werden. Im Gegensatz zu klassischen Verfahren nennt man dieses Verfahren semi-iterative. Zur Bestimmung von  $\omega^\kappa$  gibt es mehrere Möglichkeiten. Eine davon ist die Anwendung des Verfahrens der *konjugierten Gradienten (CG)* [34], welches selbstadaptiv ist, d. h.  $\omega^\kappa$  läßt sich im Verlauf der Iteration vom Algorithmus selbst bestimmen. Nachteilig ist jedoch, daß das Verfahren nur auf lineare Systeme mit Koeffizientenmatrix anwendbar ist, welche symmetrisch und positiv definit ist.

Als Alternative dazu wurden unlängst zahlreiche Varianten, wie zum Beispiel *GCR* [10], *ORTHOMIN* [10], *ORTHODIR* [10], *GMRES* [64], *CGS* [72], *Bi-CGSTAB* [80] usw. für unsymmetrische Systeme, wie das Gleichungssystem (4.5), vorgeschlagen. Die Leistungsfähigkeit derartiger Verfahren hängt stark von der Eigenwertverteilung und der Konditionszahl der Koeffizientenmatrix  $[A]^\nu$  ab. Man hat also mit Konvergenzschwierigkeiten bei den ungünstigen Eigenschaften von  $[A]^\nu$  zu rechnen. Durch eine geeignete Transformation des Gleichungssystems kann man aber die Eigenschaften der Koeffizientenmatrix und damit die Konvergenzgeschwindigkeit verbessern.

Unter der Annahme, daß sich die Zerlegungsmatrix  $[K]^\nu$  eines klassischen Iterationsverfahrens wie

$$[K]^\nu = [K_L]^\nu [K_R]^\nu \quad (4.32)$$

faktorisieren läßt, erhält man das *vorkonditionierte System*

$$[K_L^{-1}]^\nu [A]^\nu [K_R^{-1}]^\nu [K_R]^\nu \Delta \mathbf{Q}^\nu = [K_L^{-1}]^\nu \mathbf{RHS}(\mathbf{Q}^\nu) \quad (4.33)$$

für das Gleichungssystem (4.5). Unter der *Vorkonditionierung* versteht man, daß ein Gleichungssystem durch ein noch leichter zu lösendes Gleichungssystem mit derselben Lösung ersetzt wird. Die Zerlegungsmatrix  $[K]^\nu$  wird auch als *Vorkonditionierer* bezeichnet.

Der Vorkonditionierer sollte schwach besetzt, leicht zu faktorisieren und eine gute Approximation an  $[A]^\nu$  sein, d. h.  $[K^{-1}]^\nu [A]^\nu \approx [I]$  besitzen. Die Verfahren ohne Vorkonditionierung sind kaum leistungsfähig. Entsprechend einer geeigneten Wahl von  $[K_L^{-1}]^\nu$  und  $[K_R^{-1}]^\nu$  sind noch folgende Vorkonditionierungen möglich:

$$[K^{-1}]^\nu [A]^\nu \Delta \mathbf{Q}^\nu = [K^{-1}]^\nu \mathbf{RHS}(\mathbf{Q}^\nu), \quad ([K_R]^\nu = I, [K_L]^\nu = [K]^\nu) \quad (4.34)$$

und

$$[A]^\nu [K^{-1}]^\nu [K]^\nu \Delta \mathbf{Q}^\nu = \mathbf{RHS}(\mathbf{Q}^\nu), \quad ([K_L]^\nu = I, [K_R]^\nu = [K]^\nu). \quad (4.35)$$

Ein vorkonditioniertes *CG*-Typ-Verfahren erhält man, indem  $[A]^\nu$  durch  $[K_L^{-1}]^\nu [A]^\nu [K_R^{-1}]^\nu$ ,  $\Delta \mathbf{Q}^\nu$  durch  $[K_R]^\nu \Delta \mathbf{Q}^\nu$  und  $\mathbf{RHS}(\mathbf{Q}^\nu)$  durch  $[K_L^{-1}]^\nu \mathbf{RHS}(\mathbf{Q}^\nu)$  in dem verwendeten Algorithmus ersetzt wird.

Da jedes *CG*-Typ-Verfahren eigene Vor- und Nachteile bezüglich des Speicherplatzbedarfs und des Rechenaufwands bringt, ist ein *CG*-Typ-Verfahren in verschiedener Hinsicht vor einem andern nicht absolut vorzüglich. Also wurde in dieser Arbeit das *Bi-CGSTAB*-Verfahren nach *van der Vorst* [80], ein *Lanczos*-Typ-Verfahren, gewählt, welches eine stabilisierte Variante des *CGS*-Verfahrens nach *Sonneveld* [72] ist. Die geeignete Wahl der Vorkonditionierung spielt eine entscheidende Rolle für die Leistungsfähigkeit des verwendeten Verfahrens [4]. Nach der Erfahrung haben sich insgesamt die *symmetrische Gauss-Seidel*-Typ-Verfahren und das *unvollständige Punkt-LU-Zerlegung*-Verfahren zusammen mit der Vorkonditionierung (4.35) als leistungsfähig bewährt [25]. Dies wurde auch in [11], [86] festgestellt. Eine allgemeine Form des *vorkonditionierten Bi-CGSTAB*-Verfahrens für das Gleichungssystem (4.5) lautet wie folgt:

$$\mathbf{r}^{\kappa=0} = \mathbf{RHS}(\mathbf{Q}^\nu) - [A]^\nu \Delta \mathbf{Q}^{\nu, \kappa=0},$$

$$\hat{\mathbf{r}}^{\kappa=0} \text{ ist ein beliebiger Vektor, so daß } (\hat{\mathbf{r}}^{\kappa=0}, \mathbf{r}^{\kappa=0}) \neq 0, \text{ z.B. } \hat{\mathbf{r}}^{\kappa=0} = \mathbf{r}^{\kappa=0},$$

$$\rho^{\kappa=0} = \alpha = \omega^{\kappa=0} = 1,$$

$$\mathbf{v}^{\kappa=0} = \mathbf{p}^{\kappa=0} = 0,$$

$$\begin{aligned}
\rho^{\kappa+1} &= (\hat{\mathbf{r}}^0, \mathbf{r}^\kappa), \\
\beta &= (\rho^{\kappa+1}/\rho^\kappa)(\alpha/\omega^\kappa), \\
\mathbf{p}^{\kappa+1} &= \mathbf{r}^\kappa + \beta(\mathbf{p}^\kappa - \omega^\kappa \mathbf{v}^\kappa), \\
\mathbf{v}^{\kappa+1} &= [K_L^{-1}]^\nu [A]^\nu [K_R^{-1}]^\nu \mathbf{p}^{\kappa+1}, \\
\alpha &= \rho^{\kappa+1}/(\hat{\mathbf{r}}^0, \mathbf{v}^{\kappa+1}), \\
\mathbf{s} &= \mathbf{r}^\kappa - \alpha \mathbf{v}^{\kappa+1}, \\
\mathbf{t} &= [K_L^{-1}]^\nu [A]^\nu [K_R^{-1}]^\nu \mathbf{s}, \\
\omega^{\kappa+1} &= (\mathbf{t}, \mathbf{s})(\mathbf{t}, \mathbf{t}), \\
\Delta \mathbf{Q}^{\nu, \kappa+1} &= \Delta \mathbf{Q}^{\nu, \kappa} + [K_R^{-1}]^\nu (\alpha \mathbf{p}^{\kappa+1} + \omega^{\kappa+1} \mathbf{s}), \\
\mathbf{r}^{\kappa+1} &= \mathbf{s} - \omega^{\kappa+1} \mathbf{t}.
\end{aligned} \tag{4.36}$$

In diesem vorkonditionierten Algorithmus tauchen  $[K_R^{-1}]^\nu \mathbf{p}^{\kappa+1}$  und  $[K_R^{-1}]^\nu \mathbf{s}$  jeweils zweimal auf. Dabei können die Matrix-Vektor-Multiplikationen nur einmal vorgenommen werden. Jedoch erfordert dies Speicherplätze für zwei Vektoren.

### 4.1.3 Eingefrorene Funktionalmatrix

Die Funktionalmatrix beim *Newton-Raphson*-Verfahren wird nur benutzt, um den Korrekturvektor zu berechnen. Man möchte gerne das Verfahren vereinfachen, gleichzeitig aber die schnelle Konvergenz des Verfahrens erhalten. Es gibt eine Reihe modifizierter *Newton-Raphson*-Verfahren [6]. Die Auswertung und Faktorisierung der Funktionalmatrix bei jedem Iterationsschritt kann viel Rechenzeit kosten. Eine populäre Technik zur Verbesserung der Gesamtleistung für das *Newton-Raphson*-Verfahren ist es, die Funktionalmatrix zu Beginn der *Newton-Raphson*-Iteration nur einmal auszuwerten und zu faktorisieren, und dann die so faktorisierte Matrix im Laufe der *Newton-Raphson*-Iteration wiederholt zu benutzen. Obwohl diese Technik die Konvergenzrate vermindern kann, steigert sie die Effizienz, wenn sich die Funktionalmatrix mit der Iteration nicht rasch ändern sollte, d. h. eine gute Anfangslösung vorgeschrieben ist.

Diese Idee könnte auch auf das Lösungsverfahren (4.5) und (4.6) anwendbar sein, was in

$$[A]^{\nu=0} \Delta \mathbf{Q}^\nu = \mathbf{RHS}(\mathbf{Q}^\nu) \tag{4.37}$$

mit

$$\mathbf{Q}^{\nu+1} = \mathbf{Q}^\nu + \zeta \Delta \mathbf{Q}^\nu \quad (4.38)$$

resultiert. Eine Reihe linearer Gleichungssysteme werden also mit derselben Koeffizientenmatrix für die unterschiedlichen rechten Seiten aufgelöst. Die Diagonalmatrix  $[D]^{\nu=0}$ , welche sich auf ein lineares iteratives Verfahren bezieht, wird nur einmal bei jedem Zeitschritt faktorisiert, so daß die faktorisierte Matrix  $[D]^{\nu=0}$ , wenn man sie abspeichert, für weitere Iterationsschritte  $\nu$  gemeinsam benutzt werden kann. Dieses modifizierte Verfahren kann gewinnbringend eingesetzt werden, um den Nachteil impliziter Verfahren abzumildern, bei jedem Iterationsschritt  $\nu$  ein lineares Gleichungssystem lösen zu müssen. Falls sich das Konvergenzverhalten verschlechtert, kann man von der intermittierend, aber nicht bei jedem Iterationsschritt  $\nu$  ausgewerteten Funktionalmatrix Gebrauch machen.

## 4.2 Iteratives Mehrgitterverfahren

Es ist bekannt, daß klassische Iterationsverfahren für die ersten Iterationen schnell und danach sehr langsam konvergieren. Sei etwa  $\tilde{\mathbf{x}}$  eine Näherung an die Lösung  $\mathbf{x}$  des algebraischen Gleichungssystems, welches sich aus einer Diskretisierung linearer oder nichtlinearer elliptischer Differentialgleichungen ergibt, so sollte ein Iterationsverfahren den Fehlervektor

$$\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}} \quad (4.39)$$

in allen Komponenten möglichst stark und gleichmäßig reduzieren. Dies ist aber im allgemeinen nicht der Fall. Diese Fehlerreduktion ist vielmehr umso schlechter, je weniger sich die Gitterwerte des Fehlervektors in benachbarten Knoten unterscheiden, d. h. je glatter der Fehlervektor ist. Zerlegt man den Fehlervektor  $\mathbf{e}$  durch *diskrete Fouriertransformation* [19] in seine verschiedenen Frequenzanteile, so erkennt man, daß die meisten Iterationsverfahren die langwelligen Anteile des Fehlervektors schlecht und die kurzwelligen Anteile sehr gut dämpfen. Das heißt, daß die Fehler in der Anfangsphase sehr schnell durch ein Iterationsverfahren gedämpft werden. Überträgt man die nach einigen Iterationen gewonnene langwellige Gitterfunktion (ein Gleichungssystem mit den langwelligen Fehleranteilen) auf das nächstgrößere Gitter (indem man beispielsweise nur jeden zweiten Knoten des nächstfeineren Gitters betrachtet), so entsteht eine höherfrequente Funktion, welche durch das Iterationsverfahren schneller als die ursprüngliche gedämpft wird.

Die dem iterativen Mehrgitterverfahren zugrundeliegende Idee besteht nun darin, die Fehleranteile verschiedener Frequenzen auf mehreren, hintereinander geschachtelten Gittern zu dämpfen. Auf den feineren Gittern wird das Iterationsverfahren eigentlich nur als Fehlerglättung verwendet, um die hochfrequenten, auf den gröberen Gittern nicht sichtbaren Anteile zu dämpfen. Da die Konvergenzgeschwindigkeit klassischer Iterationsverfahren offenbar durch die niedrigsten zu betrachtenden Frequenzen bestimmt wird, kann man hoffen, auf diese Art schnelle Verfahren konstruieren zu können.

Mehrgitteriterationen werden auf einer Reihe von Gittern  $G_l (l = 1, 2, 3, \dots, L)$  vorgenommen, welche ineinander geschachtelt sind. Darin bringt  $G_L$  das feinste Gitter zum Ausdruck, auf dem ein Lösungsvektor gesucht ist, und die Gitter werden immer gröber, je kleiner  $l$  wird. Die

gröberen Gitter werden für strukturierte Gitter oft durch Entfernung jeder zweiten Gitterlinie aus dem feineren Gitter in jeder Gitterrichtung erzeugt.

Iterative Mehrgitterverfahren wurden ursprünglich zur Lösung diskreter Differenzgleichungen vorgeschlagen, welche sich aus einer Diskretisierung elliptischer Differentialgleichungen ergeben. Diese Lösungsverfahren können auch auf Anfangs-Randwertprobleme angewendet werden, um eine Folge algebraischer Gleichungssysteme zu lösen, welche bei einer impliziten Zeitdiskretisierung entstehen. Insbesondere bei der Anwendung iterativer Mehrgitterverfahren auf das algebraische nichtlineare Gleichungssystem (3.39) unterscheidet man zwischen linearen und nichtlinearen Verfahren. Lineare Mehrgitterverfahren werden zur Lösung einer Folge linearer Gleichungssysteme in der linearisierten Form (4.5) verwendet, während nichtlineare Mehrgitterverfahren direkt auf das algebraische nichtlineare Gleichungssystem (3.39) angewendet werden. In dieser Arbeit werden die zwei iterativen Mehrgitterverfahren zur Lösung des algebraischen nichtlinearen Gleichungssystems (3.39) zu jedem diskreten Zeitpunkt  $t^{n+m}$  betrachtet.

### 4.2.1 Lineares Verfahren

Wie bereits erwähnt, hängt die Konvergenzgeschwindigkeit des Lösungsverfahrens (4.5) und (4.6) von der Leistungsfähigkeit des verwendeten Lösungsverfahrens für eine Folge linearer Systeme ab, so daß der Rückgang der Konvergenzrate des linearen Lösungsverfahrens bei großen und nicht stark blockdiagonaldominanten Systemen einen erheblichen Rechenaufwand in impliziten Verfahren mit sich bringen kann. Ist

$$[A]_l^\nu \mathbf{x}_l = \mathbf{rhs}_l \quad ([A]_{l=L}^\nu = [A]^\nu, \quad \mathbf{x}_{l=L} = \Delta \mathbf{Q}^\nu, \quad \mathbf{rhs}_{l=L} = \mathbf{RHS}(\mathbf{Q}^\nu)) \quad (4.40)$$

ein lineares Gleichungssystem auf einem Gitter  $G_l$  und soll  $\tilde{\mathbf{x}}_l^{alt}$  eine nach einigen Iterationen gewonnene Näherung für die Lösung des Gleichungssystems sein, so erhält man das *Defektkorrektursystem*

$$[A]_l^\nu \mathbf{e}_l = \mathbf{rhs}_l - [A]_l^\nu \tilde{\mathbf{x}}_l^{alt} = \mathbf{res}_l, \quad (4.41)$$

wobei  $\mathbf{e}_l = \mathbf{x}_l - \tilde{\mathbf{x}}_l^{alt}$  der *Lösungsfehler* ist. Aus der exakten Lösung  $\mathbf{e}_l$  des Systems (4.41) könnte man  $\mathbf{x}_l$

$$\mathbf{x}_l = \tilde{\mathbf{x}}_l^{alt} + \mathbf{e}_l \quad (4.42)$$

berechnen.

Da es sich bei  $\mathbf{e}_l$  jedoch um eine glatte Gitterfunktion handelt, ist es sinnvoll, statt des Systems (4.41) das ähnliche, aber kleinere System

$$[A]_{l-1}^\nu \mathbf{e}_{l-1} = I_l^{l-1} \mathbf{res}_l = \mathbf{rhs}_{l-1} \quad (4.43)$$

auf einem gröberen Gitter  $G_{l-1}$  näherungsweise zu lösen. Dabei ist  $I_l^{l-1}$  ein Operator für die Übertragung des Defekts  $\mathbf{res}_l$  von  $G_l$  auf  $G_{l-1}$  und  $[A]_{l-1}^\nu$  die Koeffizientenmatrix auf dem

Gitter  $G_{l-1}$ . Für die Wahl von  $I_l^{l-1}$  gibt es mehrere Möglichkeiten. Da man bei knotenzentrierten Differenzen einfache algebraische Operatoren sinnvoll verwenden kann, wird der Defekt im inneren Rechengebiet durch die *Neun-Punkt-Restriktion*, am Rand durch die *Injektion*

$$I_l^{l-1} \mathbf{res}_l = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \mathbf{res}_l, \quad I_l^{l-1} \mathbf{res}_l = \mathbf{res}_l \quad (4.44)$$

übertragen.

Für die Grobgitterapproximation gibt es zwei Möglichkeiten: die *Galerkin-Approximation* und die *Grobgitterdiskretisierung*. Welche von beiden sinnvoller ist, kommt auf die Umstände an. Bei der *Galerkin-Approximation* kann man Mehrgitterverfahren als ein unabhängiges Lösungsverfahren in einem impliziten Verfahren benutzen. Diese Approximation ist jedoch auf lineare Gleichungssysteme beschränkt und könnte zu einer schlechten Konvergenz führen, wenn die Approximation nicht genau sein sollte. In dieser Arbeit wird die Koeffizientenmatrix  $[A]_{l-1}^\nu$  auf  $G_{l-1}$  durch die Injektion von  $\mathbf{Q}_l^\nu$  zum diskreten Operator  $\tilde{\mathbf{Res}}(\mathbf{Q})$  auf  $G_{l-1}$  näherungsweise erzeugt.

Aus (4.41) geht hervor, daß die iterative Lösung des originalen Systems (4.40) mit einer beliebigen Ausgangsnäherung identisch mit der iterativen Lösung des Defektkorrektursystems (4.41) mit der Ausgangsnäherung  $\mathbf{e}_l=0$  ist. In diesem Zusammenhang ist wichtig, daß man die iterative Lösung von (4.43) mit der Ausgangsnäherung  $\mathbf{e}_{l-1}=0$  beginnt. Angenommen, man hat das System (4.43) genügend gut gelöst und einen angenäherte Lösungsfehler  $\tilde{\mathbf{e}}_{l-1}$  erhalten, so wird der Fehler durch einen Operator auf  $G_l$  wieder interpoliert. Dies geschieht in dieser Arbeit durch die *stückweise lineare Interpolation*

$$I_{l-1}^l \tilde{\mathbf{e}}_{l-1} = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tilde{\mathbf{e}}_{l-1}. \quad (4.45)$$

Aus (4.42) berechnet sich dann auf  $G_l$  ein neuer Wert

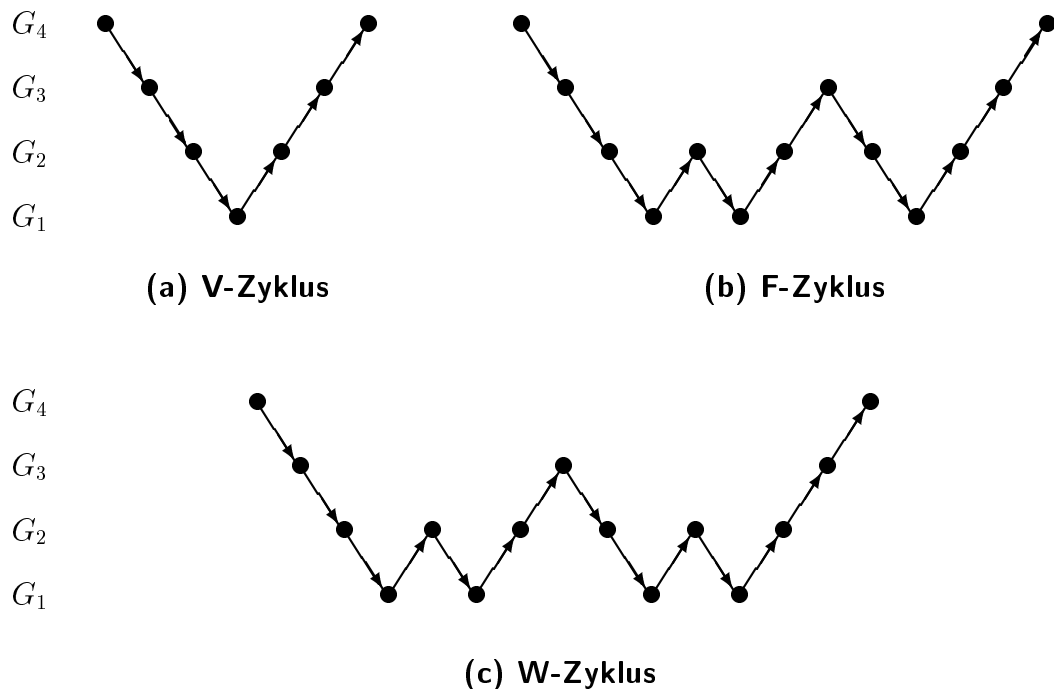
$$\tilde{\mathbf{x}}_l^{neu} = \tilde{\mathbf{x}}_l^{alt} + \omega_l I_{l-1}^l \tilde{\mathbf{e}}_{l-1}, \quad (4.46)$$

wobei  $\omega_l$  ein Faktor

$$\omega_l = \frac{(\mathbf{rhs}_{l-1}, \tilde{\mathbf{e}}_{l-1})}{([A]_{l-1}^\nu \tilde{\mathbf{e}}_{l-1}, \tilde{\mathbf{e}}_{l-1})} \quad (4.47)$$

zur Minimierung des Lösungsfehlers ist [40]. Bei der Interpolation von  $\tilde{\mathbf{e}}_{l-1}$  entstehen neue hochfrequente Fehleranteile, welche durch anschließende Nachiterationen mit  $\tilde{\mathbf{x}}_l^{neu}$  als Ausgangsnäherung auf  $G_l$  gedämpft werden. Dieser Prozeß ist rekursiv.

Wenn die Konvergenz für (4.43) langsamer wird oder  $G_{l-1}$  groß ist, also die Lösung einen hohen Rechenaufwand erfordert, dann wird der Defekt  $\mathbf{res}_{l-1}$ , welcher auf  $G_{l-1}$  nach einigen Iterationen erzielt wurde, auf  $G_{l-2}$  übertragen. Auf dem größten Gitter  $G_1$  berechnet man eine Näherung an die Lösung des Defektkorrektursystems (4.41) entweder direkt oder iterativ. Den soweit geschilderten Gitterwechsel bezeichnet man als *V-Zyklus*, welcher vom feinsten Gitter  $G_L$  aus direkt zum größten Gitter  $G_1$  und von dort wieder direkt zum feinsten geht.



- Durchführung der Iteration auf dem jeweiligen Gitter
- ↘ Abstieg auf das nächstgrößere Gitter durch Bilden des Systems (4.43)
- ↗ Aufstieg auf das nächstfeinere Gitter gemäß (4.46)

**Bild 4. 1: Verschiedene Zyklusdiagramme für den Gitterwechsel**

Wann den Gitterwechsel vorzunehmen ist, kann man entweder vor Beginn der Rechnung festlegen (*starre* Technik) oder im Verlauf der Rechnung anhand der erzielten Resultate entscheiden (*adaptive* Technik) [19]. Bei der adaptiven Technik hält man sich daran, ob für den Wechsel auf das nächstgrößere Gitter die hochfrequenten Fehleranteile auf dem feinen Gitter genügend klein, aber noch niederfrequente Anteile vorhanden sind, und für den Wechsel auf das nächstfeinere Gitter die Gleichungen auf dem groben Gitter genügend gelöst sind, also alle Fehleranteile auf dem groben Gitter klein sind. Es ist aber nicht einfach zu beurteilen, ob das vorgegebene Kriterium für einen Gitterwechsel im Verlauf der Rechnung erfüllt ist. Bei der starren Technik ist zwar nicht sicherzustellen, daß man das voreingestellte Kriterium für einen Gitterwechsel immer trifft, dafür ist diese Technik jedoch robust im Konvergenzverhalten und einfach zu programmieren. Daher wird in dieser Arbeit eine starre Technik eingesetzt. Die Anzahl der Vor- und Nachiteration richtet sich nach der Regel  $2^{L-l+ic}$  ( $ic = 1, 2, 3..$ ). Die Vorgabe der Nummer  $ic$  ist problemabhängig. Für den Gitterwechsel zwischen Fein- und Grobgittern gibt es einige Möglichkeiten. Die oft verwendeten Gitterwechsel mit vier Gitterebenen sind in Bild 4.1 dargestellt, wobei jedoch viele Varianten möglich sind.



Da bei dem Abstiegsprozeß auf jedem Gitter  $G_l$  die Koeffizientenmatrix erzeugt, aber auch die Diagonalmatrix  $[D]^\nu$  faktorisiert wird, werden beide auch beim Aufstiegsprozeß und weiterhin für den nächsten Zyklus in einem nichtlinearen Iterationsschritt  $\nu$  unverändert ausgenutzt. Setzt man die eingefrorene Koeffizientenmatrix  $[A]_l^{\nu=0}$  für das Mehrgitterverfahren ein, werden die Auswertung der Koeffizientenmatrix und die Faktorisierung der Diagonalmatrix  $[D]^\nu$  auf jedem Gitter  $G_l$  nur einmal für einen Zeitschritt vorgenommen. Die Wahl des Iterationsverfahrens übt einen entscheidenden Einfluß auf die Leistungsfähigkeit des Mehrgitterverfahrens aus, ist aber problemabhängig. Die Rolle der Iteration im Mehrgitterverfahren besteht darin, die hochfrequenten Anteile des Lösungsfehlers schnell zu dämpfen. Somit sollte ein Iterationsverfahren eher ein Glätter als Löser sein. Insgesamt gesehen hat sich für diese Arbeit das Verfahren der *unvollständigen Punkt-Zerlegung* (4.25) mit der eingefrorenen Koeffizientenmatrix  $[A]_l^{\nu=0}$  bewährt [26]. Nur ein Zyklus mit einer Iteration auf jedem Gitter wird durchlaufen, da das Lösungsverfahren (4.5) und (4.6) keine gute Näherung an  $\Delta \mathbf{Q}^\nu$  erfordert.

Das bisher dargestellte Mehrgitterverfahren bezeichnet man als *Correction-Storage*-Schema. Diese Mehrgittertechnik ist auf lineare Probleme begrenzt, weil die Beziehung (4.41) nur für derartige Probleme gültig ist. Im Gegensatz dazu approximiert das sogenannte *Full-Approximation-Storage*-Schema, welches nachfolgend erläutert werden soll, statt von  $\mathbf{e}_{l-1}$   $\mathbf{x}_{l-1} = \tilde{\mathbf{x}}_{l-1}^{alt} + \mathbf{e}_{l-1}$  auf  $G_{l-1}$ , so daß dieses Schema unabhängig von der Linearität anwendbar ist. Allerdings ist dieses Verfahren noch etwas aufwendig, wenn es auf lineare Probleme Anwendung findet. Das Gleichungssystem (4.11) für stationäre Strömungen kann durch Einsatz der Mehrgittertechnik bei jedem Zeitschritt effektiv gelöst werden.

## 4.2.2 Nichtlineares Verfahren

Die Gleichung (3.39) stellt ein algebraisches nichtlineares Gleichungssystem für den Lösungsvektor  $\mathbf{Q}^{n+m}$  zum diskreten Zeitpunkt  $t^{n+m}$  dar, so daß für die direkte Anwendung der Mehrgittertechnik auf diese Gleichung auf das *Full-Approximation-Storage*-Schema, eine Verallgemeinerung des *Correction-Storage*-Verfahrens, zurückgegriffen werden kann. Bei der Anwendung des Verfahrens auf (3.39) und aus Übersichtsgründen unter Vernachlässigung des Zeitschrittzählers  $n + m$  erhält man auf einer Gitterebene  $G_l$  ein modifiziertes Gleichungssystem der Form

$$\frac{V}{\Delta t} \alpha^m \mathbf{Q}_l - \mathbf{N}_l(\mathbf{Q}_l) = \tau_l^L \quad (\tau_{l=L}^L = 0), \quad (4.48)$$

wobei  $\tau_l^L$  der lokale Diskretisierungsfehler auf  $G_l$  relativ zu  $G_L$  ist. Die Durchführung einiger nichtlinearer Iterationen nach dem iterativen Prozeß (4.5) und (4.6) ergibt eine Näherung  $\tilde{\mathbf{Q}}_l^{alt}$  an  $\mathbf{Q}_l$ , und der entsprechende Defekt  $\mathbf{d}_l$  lautet:

$$\mathbf{d}_l = \tau_l^L + \mathbf{N}_l(\tilde{\mathbf{Q}}_l^{alt}) - \frac{V}{\Delta t} \alpha^m \tilde{\mathbf{Q}}_l^{alt}. \quad (4.49)$$

Im Gegensatz zum *Correction-Storage*-Schema wird beim *Full-Approximation-Storage*-Schema der Lösungsvektor  $\mathbf{Q}_{l-1}$  auf  $G_{l-1}$  approximiert, indem man das Grobgitterproblem

$$\frac{V}{\Delta t} \alpha^m \mathbf{Q}_{l-1} - \mathbf{N}_{l-1}(\mathbf{Q}_{l-1}) = \tau_{l-1}^L \quad (4.50)$$

mit

$$\tau_{l-1}^L = \frac{V}{\Delta t} \alpha^m I_l^{l-1} \tilde{\mathbf{Q}}_l^{alt} - \mathbf{N}_{l-1}(I_l^{l-1} \tilde{\mathbf{Q}}_l^{alt}) + \sigma I_l^{l-1} \mathbf{d}_l \quad (4.51)$$

auf  $G_{l-1}$  löst, wobei  $\sigma \in (0, \infty)$  ein positiver Parameter zur Verbesserung der Grobgitterapproximation zum Feingitterproblem (4.48) ist [40]. Für die Anfangsnäherung in (4.50) kann  $I_l^{l-1} \tilde{\mathbf{Q}}_l^{alt}$ , aber auch die zuletzt erzielte Approximation an  $\mathbf{Q}_{l-1}$  zur Verwendung kommen. Der Grobgitteroperator  $\mathbf{N}_{l-1}$  wird in gleicher Weise wie  $\mathbf{N}_L$  konstruiert. Nach der angenäherten Lösung von (4.50) wird eine Näherung  $\tilde{\mathbf{Q}}_{l-1} - I_l^{l-1} \tilde{\mathbf{Q}}_l^{alt}$  an die Korrektur  $\mathbf{Q}_{l-1} - I_l^{l-1} \tilde{\mathbf{Q}}_l^{alt}$  auf  $G_l$  übertragen, mit deren Hilfe die Feingitterapproximation  $\tilde{\mathbf{Q}}_l^{alt}$  durch die neue Approximation

$$\tilde{\mathbf{Q}}_l^{neu} = \tilde{\mathbf{Q}}_l^{alt} + \omega I_{l-1}^l (\tilde{\mathbf{Q}}_{l-1} - I_l^{l-1} \tilde{\mathbf{Q}}_l^{alt}) / \sigma, \quad (4.52)$$

korrigiert wird, wobei  $\omega \in (0, 2)$  auch ein Faktor zur Verbesserung der Korrektur ist [40].

Dann werden einige Iterationen für (4.48) mit der besseren Anfangsnäherung  $\tilde{\mathbf{Q}}_l^{neu}$  durchgeführt. Wenn die Näherung  $\tilde{\mathbf{Q}}_L$  auf  $G_L$  ein vorgeschriebenes Konvergenzkriterium nicht erfüllt, wird der gleiche Prozeß wiederholt. Die Randwerte werden nicht auf Grob- oder Feingitter übertragen, d. h. die Randwerte werden auf jedem Gitter so berechnet, daß die gegebenen Randbedingungen erfüllt werden können. Die Mechanismen für die Übertragung von Informationen zwischen Gitterebenen erfolgen in gleicher Weise wie bei dem *Correction-Storage*-Schema. Diese nichtlineare Mehrgittertechnik kann auch für stationäre Strömungen angewendet werden, wobei das nichtiterative Zeitschrittverfahren (4.11), (4.12) als Glättungsiteration eingesetzt wird.

Bisher wurden zwei Mehrgitterverfahren zur Lösung des algebraischen nichtlinearen Gleichungssystems (3.39) in impliziten Verfahren für instationäre Strömungen erläutert. Die Rolle der Iteration im Mehrgitterverfahren besteht darin, die hochfrequenten Anteile des Lösungsfehlers so schnell wie möglich zu dämpfen. Es ist bekannt [7], daß Relaxationsverfahren wie (4.15) die hochfrequenten Fehler gut glätten. Nach dem rechnerischen Experiment hat es sich gezeigt, daß für das nichtlineare Gleichungssystem (3.39) die lineare Mehrgittertechnik mittels (4.15) leistungsfähiger ist als die nichtlineare Mehrgittertechnik mittels (4.2).