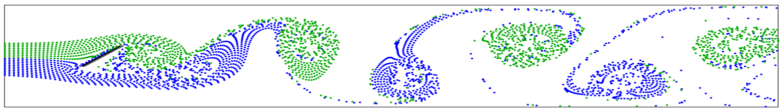
**Institut für Angewandte Mathematik der Universität Bonn**
Abteilung Wissenschaftliches Rechnen und Numerische Simulation
Professor Dr. Michael Griebel

**Praktikum Sommersemester 2001**

# Numerische Simulation und Visualisierung

- Leitung: *Prof. Dr. M. Griebel*, Zi. 604, Tel. 73 3437
- Mitarbeiter: *F. Koster*, Wegelerstr. 4, Tel. 73 3173



## Blatt 4

## 1. Freie Randwertprobleme

Nachdem wir bisher Strömungen nur in fest vorgegebenen Gebieten berechnet haben, wollen wir jetzt sogenannte „freie Randwertprobleme“ behandeln. Das sind Probleme, bei denen sich die Form des Gebietes, in dem sich das Fluid befindet, zeitabhängig verändern kann. Ein Beispiel dafür ist der in ein mit Flüssigkeit gefülltes Becken fallende Tropfen, der auf dem ersten Aufgabenblatt auf der ersten Seite unten dargestellt ist. Ein weiteres Beispiel ist ein Damm, bei dem die Mauer auf einen Schlag entfernt wird (d.h. der Damm bricht) und die Flüssigkeit sich ausbreiten kann.
Bevor der Druck und die Geschwindigkeiten zu einem festen Zeitpunkt mit Hilfe der diskretisierten Navier–Stokes–Gleichungen berechnet werden können, muß zunächst das aktuelle Gebiet bestimmt werden. Dies geschieht durch die Verwendung von Partikeln, die von Zeitschritt zu Zeitschritt fortbewegt werden und mit deren Hilfe die Zellen des “staggered grid”, die nicht Hinderniszellen sind,

| | |
|---|---|
| •in leere Zellen: | Zellen ohne Partikel bzw. ohne Flüssigkeit, |
| •in innere Zellen: | Zellen, die Partikel enthalten und nicht an *leere Zellen* angrenzen und |
| •in Oberflächenzellen: | Zellen, die Partikel enthalten aber auch *leeren Zellen* benachbart sind |

aufgeteilt werden. Die Oberflächenzellen bilden dann den freien Rand, im Gegensatz zum festen Rand des rechteckigen Grundgebietes $\Omega$. Zusätzlich müssen am freien Rand neue physikalische Randbedingungen berücksichtigt werden.

## 2. Die Bestimmung der Gebietsform

Zur Bestimmung des Gebietes, das in einem Zeitschritt behandelt wird, verwenden wir Partikel, die in den `Particlelines` von Blatt 2 abgespeichert werden können.

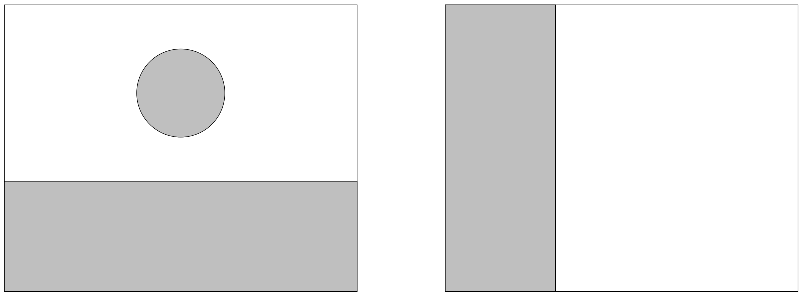


Abb. 1: Ausgangskonfiguration “Fallender Tropfen” und “Brechender Damm”

Zu Beginn muß die Ausgangsgeometrie des mit Flüssigkeit gefüllten Gebietes gekennzeichnet werden (siehe auch Abbildung 1).
Dazu wird jeder Zelle des Gitters eine feste Anzahl potentieller Partikelpositionen zugeordnet (meist 9 oder 16). Ein Partikel wird jedoch nur dann auf seinen Platz in der Zelle gesetzt, d.h. in eine Liste von `Particlelines` aufgenommen, wenn seine Position innerhalb des zu beschreibenden Fluid–Gebietes liegt (siehe Abbildung 2).

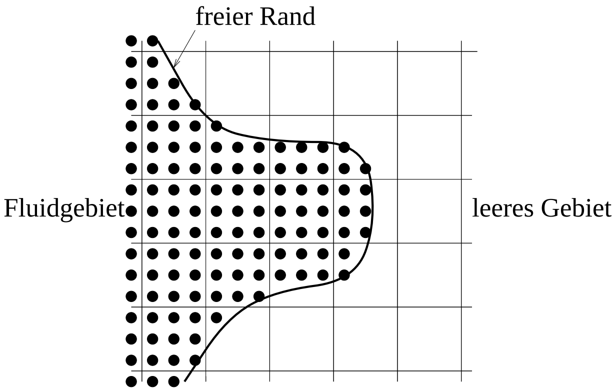



Abb. 2: Partikelpositionen in einer Anfangskonfiguration

Wenn verschiedene Bereiche in der Ausgangsgeometrie unterschieden werden sollen (beim fallenden Tropfen z.B. Tropfen und Becken), können die zugehörigen Partikel in unterschiedliche Listen des Feldes `Particlelines` geschrieben werden.
Im Zeitschritt $n$ seien die Partikelpositionen zum Zeitschritt $t_n$ bekannt. Diese aktuellen Positionen lassen sich nun benützen, um festzustellen, ob eine Zelle leer ist, sich ganz im Flüssigkeitsgebiet befindet, oder auf dessen Rand liegt. Dann können entlang des freien Randes die zugehörigen Randwerte bestimmt werden, bevor im Inneren des Flüssigkeitsgebietes die Druck– und Geschwindigkeitswerte berechnet werden. Die Werte außerhalb des Flüssigkeitsgebietes sind nicht von Interesse. Schließlich müssen am Ende des Zeitschritts noch mit den neu berechneten Geschwindigkeiten die Partikelpositionen zum Zeitpunkt $t_{n+1}$ ermittelt werden.

Zum Markieren der Zellen kann das Integer-Feld `int **FLAG`, das bereits in Blatt 2 eingeführt wurde, erweitert werden. Neben den Belegungen `C_B`, sowie `B_N,...,B_NW`[1] für die Hinderniszellen, die das Hindernisgebiet $\mathcal{H}$ approximieren, bezeichne das Makro `C_F` jetzt die *inneren Zellen* und das Makro `C_E` die *leeren Zellen*.
Wie die Menge der Randzellen der Hindernisse muß auch die Menge der Oberflächenzellen noch weiter aufgeschlüsselt werden. Dazu geben wir in dem Flag-Feld an, in welcher der vier “Himmelsrichtungen” sich jeweils leere Zellen befinden. So kann zum Beispiel eine Zelle, die nur links eine leere Zelle als Nachbarn hat, mit `C_W`, eine Zelle, deren Nachbarn unten und rechts leer sind, mit `C_SO` und eine Zelle, die nur leere Zellen als Nachbarn hat, mit `C_NSWO` codiert werden. Insgesamt sind so 15 verschiedene Typen freier Oberflächenzellen zu unterscheiden. Die Größen `C_xy` stehen dabei für Integerkonstanten, die beliebig gewählt werden können.
In einer ersten Schleife über alle Partikel sollen die Flags derjenigen Zellen mit `C_F` belegt werden, in denen sich ein Partikel befindet. Alle anderen Zellen, die nicht Hinderniszellen sind, sollen mit dem Flag `C_E` versehen werden. Gleichzeitig sollten die Partikel, die sich aus dem Grundgebiet $\mathcal{R}$ herausbewegt haben oder die in Hinderniszellen geraten sind, eliminiert werden.
In einer zweiten Schleife über alle Zellen müssen dann die Zellen, die in der ersten Schleife den Flag `C_F` erhalten haben, noch in echte innere Zellen und Oberflächenzellen aufgeteilt werden, indem festgestellt wird, ob einige, und wenn ja welche, der vier Nachbarzellen leer sind. Die Oberflächenzellen werden dabei mit dem entsprechenden Flag-Wert gekennzeichnet.

## 3. Die Randbedingungen am freien Rand

Bevor nun die Druck– und Geschwindigkeitswerte im Inneren des Flüssigkeitsgebietes berechnet werden können, müssen am freien Rand, ebenso wie am festen Rand, Randwerte bestimmt werden.

Es sind folgende Randwerte nötig: (siehe Abbildung 3):

- Druck in den Oberflächenzellen (Kreuz),
- Geschwindigkeiten an den Kanten zwischen Oberflächenzellen und leeren Zellen (dunkler Kreis),
- Geschwindigkeiten an Kanten zwischen zwei leeren Zellen, die an Ecken von Oberflächenzellen anliegen (heller Kreis).

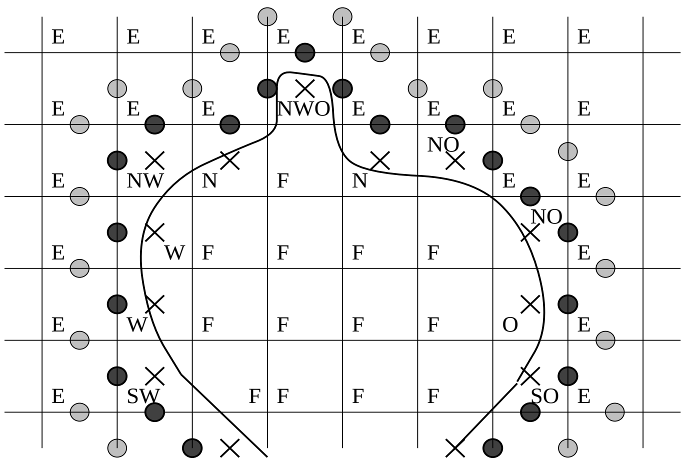


Abb. 3: Benötigte Randwerte

[1]Im Unterschied zur Definition der Randzellen in Blatt 2 sind hier natürlich all diejenigen Zellen Randzellen, die als Hinderniszellen mindestens einer inneren, leeren oder Oberflächenzelle benachbart sind.

Mit diesen Randwerten kann der Druck in den inneren Zellen mit den $F$– und $G$–Werten an den anliegenden Kanten nach Formel (11) auf Blatt 1 berechnet werden. Aus den so ermittelten Druckwerten in den inneren Zellen und den Oberflächenzellen, können gemäß den Formeln (7) und (8) auf Blatt 1 mit den entsprechenden $F$– und $G$–Werten die Geschwindigkeiten auf den Kanten bestimmt werden, die nicht an eine leere Zelle grenzen. Im Anschluß daran werden die neuen Positionen der Partikel mit der Prozedur `ADVANCE_PARTICLES` berechnet.

Da oben genannte Druck– und Geschwindigkeitswerte aus benachbarten Geschwindigkeitswerten berechnet werden (siehe unten), und im folgenden Zeitschritt alle an Oberflächenzellen angrenzende leere Zellen zu Oberflächenzellen werden können, muß auch gewährleistet sein, daß an den in Abbildung 3 mit hellen Kreisen gekennzeichneten Punkten bereits "vernünftige" Geschwindigkeitswerte existieren (da diese dann zur Bestimmung der Druckrandwerte für die Poissongleichung für den Druck benötigt werden). Außerdem werden diese Werte teilweise für die Interpolation der Geschwindigkeiten an den Partikelpositionen am Rand benötigt. Hier zeigt sich sehr deutlich die Notwendigkeit der CFL–Bedingung von Blatt 1, weil diese garantiert, daß die Partikel höchstens eine Zelle weiter wandern.

Die Randwerte werden wie folgt bestimmt:

Die physikalische Bedingung, aus der die Formeln für die Randwerte bei Vernachlässigung der Oberflächenspannung abgeleitet werden können, ist das Verschwinden der normalen und der tangentialen Komponente der Oberflächenspannung, d.h.:

$$p - \frac{2}{Re}\left(n_x\,n_x\,\frac{\partial u}{\partial x} + n_x\,n_y\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + n_y\,n_y\,\frac{\partial v}{\partial y}\right) = 0 \tag{1}$$

und

$$2\,n_x\,m_x\,\frac{\partial u}{\partial x} + (n_x\,m_y + n_y\,m_x)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + 2\,n_y\,m_y\,\frac{\partial v}{\partial y} = 0. \tag{2}$$

Dabei sind $(n_x, n_y)^T$ und $(m_x, m_y)^T$ die Einheitsvektoren, die senkrecht (nach außen zeigend) bzw. tangential zum freien Rand liegen.

Bei der Diskretisierung dieser Bedingungen müssen nun 5 Grundtypen unterschiedlicher Oberflächenzellen betrachtet werden.

1. **Zellen mit einer leeren Nachbarzelle:**
   Es wird angenommen, daß der freie Rand nahezu parallel zu den Gitterlinien verläuft, so daß entweder $n_y$ und $m_x$ oder $n_x$ und $m_y$ sehr klein sind. Gleichung (1) kann dann durch
   $$p = \frac{2}{Re}\,\frac{\partial u}{\partial x} \qquad \text{bzw.} \qquad p = \frac{2}{Re}\,\frac{\partial v}{\partial y} \tag{3}$$
   approximiert werden, wobei die erste Gleichung für senkrechte und die zweite für waagrechte Ränder gilt. Gleichung (2) wird zu
   $$\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = 0. \tag{4}$$

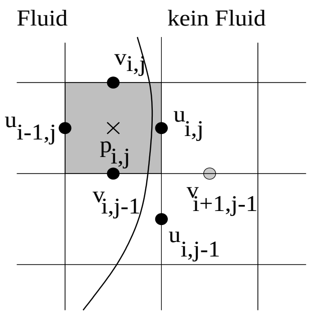



Abb. 4:
Zelle mit Flag `C_O`

Die Diskretisierungen sollen exemplarisch an Hand einer `C_O`–Zelle erläutert werden (siehe Abbildung 4):
Der Druck $p$ in der Zelle $(i, j)$ wird aus Gleichung (3) bestimmt:

$$p_{i,j} = \frac{2}{Re} \frac{(u_{i,j} - u_{i-1,j})}{\delta x}. \tag{5}$$

Zuvor wird die Geschwindigkeit $u_{i,j}$ an der Kante zwischen Oberflächen– und leerer Zelle so gewählt, daß in der Zelle $(i, j)$ die diskretisierte Version der Kontinuitätsgleichung $\partial u/\partial x + \partial v/\partial y = 0$ erfüllt ist:

$$u_{i,j} = u_{i-1,j} - \frac{\delta x}{\delta y}(v_{i,j} - v_{i,j-1}). \tag{6}$$

Zur Berechnung der Geschwindigkeit $v_{i+1,j-1}$, die einer Kante zwischen zwei leeren Zellen, die an eine Oberflächenzelle angrenzt, zugeordnet ist, wird die Bedingung (4) herangezogen:

$$v_{i+1,j-1} = v_{i,j-1} - \frac{\delta x}{\delta y}(u_{i,j} - u_{i,j-1}). \tag{7}$$

Dies darf jedoch nur geschehen, wenn die Zelle $(i{+}1,j{-}1)$ leer ist, da sonst $v_{i+1,j-1}$ auch aus der Kontinuitätsgleichung für die Zelle $(i{+}1,j{-}1)$ bestimmt werden kann. Weil zur Auswertung die Werte von $u_{i,j}$ und $u_{i,j-1}$ benötigt werden, muß dies nach der Berechnung von $u_{i,j}$ nach (6) geschehen; $u_{i,j-1}$ ist schon bekannt, da die Berechnung von links unten nach rechts oben verläuft.

Für die anderen Zellen mit *einer* benachbarten leeren Zelle (Flags `C_N,C_W,C_S`), müssen die obigen Formeln analog angewendet werden. Dabei ist darauf zu achten, daß sich zum Teil $u$ und $v$, $\delta x$ und $\delta y$ sowie Vorzeichen vertauschen. Formel (4) wird jeweils auf die links bzw. unten anliegende Kante angewendet, um alle benötigten Werte verfügbar zu haben (siehe auch Abbildung 7).

2. **Zellen mit zwei benachbarten, über Eck liegenden leeren Zellen:**
   Hier kann der freie Rand durch ein Geradenstück approximiert werden, das unter einem Winkel von 45° die Gitterlinien schneidet, so daß $n_x$, $n_y$, $m_x$ und $m_y$ vom Betrag gleich sind und sich nur durch das Vorzeichen unterscheiden.

   Formel (1) reduziert sich dann (unter Anwendung der Kontinuitätsgleichung) zu

$$p = \pm \frac{1}{Re} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \tag{8}$$

   wobei das Pluszeichen für Zellen mit Flag `C_NO` und `C_SW` und das Minuszeichen für Zellen mit Flag `C_NW` und `C_SO` gilt. Aus Formel (2) wird

$$\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} = 0. \tag{9}$$

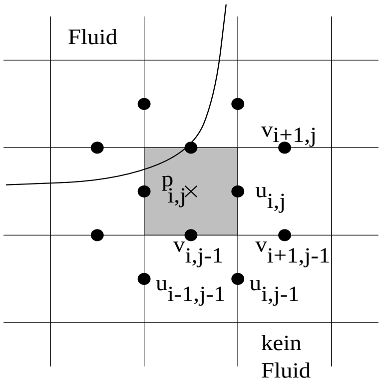



Abb. 5: Zelle mit Flag `C_NO`

Die Diskretisierungen sollen hier an Hand einer `C_SO`–Zelle erläutert werden (siehe Abbildung 5): Aus (9) und der Kontinuitätsgleichung für die Zelle $(i,j)$ folgt, daß $\partial u/\partial x = 0$ und $\partial v/\partial y = 0$. Diskretisiert wird daraus

$$u_{i,j} = u_{i-1,j}, \qquad v_{i,j-1} = v_{i,j}, \tag{10}$$

während sich der Druck in der Zelle $(i,j)$ aus Gleichung (8) durch

$$p_{i,j} = -\frac{1}{2\,Re}\left(\frac{u_{i,j+1} + u_{i-1,j+1} - u_{i,j} - u_{i-1,j}}{\delta y} + \right.\\ \left. + \frac{v_{i,j} + v_{i,j-1} - v_{i-1,j} - v_{i-1,j-1}}{\delta x}\right) \tag{11}$$

bestimmen läßt.

Da der freie Rand die linke und obere Kante nahezu unter einem Winkel von 90° durchschneidet, können wir in einer Näherung zur Berechnung von $u_{i-1,j-1}$ und $v_{i+1,j}$ Formel (4) verwenden, d.h.

$$\begin{aligned} u_{i-1,j-1} &= u_{i-1,j} + \frac{\delta y}{\delta x}(v_{i,j-1} - v_{i-1,j-1}),\\ v_{i+1,j} &= v_{i,j} - \frac{\delta x}{\delta y}(u_{i,j+1} - u_{i,j}). \end{aligned} \tag{12}$$

Dabei kann $v_{i+1,j}$ erst berechnet werden, wenn $u_{i,j+1}$ bekannt ist, d.h. $v_{i+1,j}$ wird erst bei Behandlung der Zelle $(i,j+1)$ bestimmt. Außerdem dürfen $u_{i-1,j-1}$ bzw. $v_{i+1,j}$ nur nach (12) bestimmt werden, wenn die Zelle $(i-1,j-1)$ bzw. $(i+1,j+1)$ leer ist.

Die Werte von $u_{i,j-1}$ und $v_{i+1,j-1}$ am Eck werden jeweils gleich den nächstliegendsten Werten gesetzt:

$$u_{i,j-1} = u_{i,j}, \qquad v_{i+1,j-1} = v_{i,j-1}. \tag{13}$$

Die anderen 3 Fälle mit den Flags `C_NW`, `C_NO` und `C_SW` werden analog behandelt. Dabei werden wiederum einige Vorzeichen, $\delta x$ und $\delta y$ und Indizes miteinander vertauscht. Auch hier muß darauf geachtet werden, daß die beiden Werte, bei denen die Diskretisierung von Formel (2) verwendet wird, nur gesetzt werden dürfen, wenn sie auf einer Kante auf Höhe des linken oder unteren Randes der Zelle $(i,j)$ liegen (vgl. Abbildung 7).

Bei allen Fällen muß ebenso beachtet werden, daß die entsprechenden Werte nicht auf Kanten einer Oberflächenzelle liegen dürfen.

3. **Zellen mit zwei gegenüberliegenden leeren Nachbarzellen:**
   Bei diesem und den restlichen Fällen lassen sich die Randbedingungen nicht mehr korrekt erfüllen, da entweder zwei freie Ränder in einer Zelle nicht korrekt behandelt werden können, oder da der Rand in der Zelle sehr stark gekrümmt ist und somit der Normalen– und der Tangentialvektor nicht eindeutig approximierbar sind. Bei einer genügend feinen Diskretisierung würden solche Fälle zwar nicht vorkommen, da dies aber in jedem Zeitschritt eventuell eine neue Diskretisierung erfordern würde und so die Anzahl der Unbekannten sehr groß werden könnte, müssen diese Zellen trotzdem behandelt werden.

   Wir nehmen hier an, daß sich das Fluid nur auf Grund der äußeren Kraft $g_x$ in $x$–Richtung bzw. $g_y$ in $y$–Richtung bewegt. Es ergibt sich dann für Zellen $(i,j)$ mit Flag `C_WO`
   $$u_{i,j}^{neu} = u_{i,j}^{alt} + \delta t\, g_x, \qquad u_{i-1,j}^{neu} = u_{i-1,j}^{alt} + \delta t\, g_x \tag{14}$$
   und für Zellen mit Flag `C_NS`
   $$v_{i,j}^{neu} = v_{i,j}^{alt} + \delta t\, g_y, \qquad v_{i,j-1}^{neu} = v_{i,j-1}^{alt} + \delta t\, g_y. \tag{15}$$
   Darüber hinaus werden auch hier für die in Abbildung 7 angegeben Punkte die Geschwindigkeiten aus (4) berechnet. Der Druck in der Zelle ist Null, d.h. $p_{i,j} = 0$.

4. **Zellen mit drei benachbarten leeren Zellen:**
   Die hierbei nötigen Schritte sollen am Beispiel einer Zelle mit Flag `C_NWO` veranschaulicht werden.

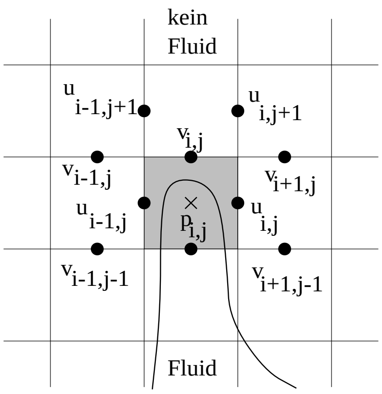



Abb. 6:
Zelle mit Flag `C_NWO`

   Die Geschwindigkeiten $u_{i-1,j}$ und $u_{i,j}$ an den beiden gegenüberliegenden Kanten, die an leere Zellen angrenzen, werden wie in (14) behandelt. Die Geschwindigkeit $v_{i,j}$ auf der dritten, an eine leere Zelle grenzende Kante, wird so gesetzt, daß die Kontinuitätsgleichung in der Zelle $(i,j)$ erfüllt ist. Die am Eck angrenzenden Geschwindigkeiten $u_{i-1,j+1}$, $u_{i,j+1}$, $v_{i-1,j}$ und $v_{i+1,j}$ werden, wie bei den Zellen mit zwei benachbarten (über Eck liegenden) leeren Zellen, mit den nächstliegenden Geschwindigkeiten, die auf Kanten der Zelle $(i,j)$ liegen ($u_{i-1,j}$, $u_{i,j}$ und $v_{i,j}$), gleichgesetzt, sofern die Zellen $(i-1,j+1)$ bzw. $(i+1,j+1)$ leer sind. Des weiteren müssen noch die Geschwindigkeiten $v_{i-1,j-1}$ und $v_{i+1,j-1}$ aus (4) bestimmt werden, wenn die Zellen $(i-1,j-1)$ bzw. $(i+1,j-1)$ leer sind. Der Druck $p_{i,j}$ ist Null.

   Die Zellen mit Flag `C_NSO`, `C_NSW` und `C_SWO` werden analog behandelt, wobei nur im Fall `C_NSO` die $v_{i-1,j-1}$ und $v_{i+1,j-1}$ entsprechenden Geschwindigkeiten berechnet werden, da nur dort alle erforderlichen Werte bereits vorhanden sind (vgl. Abbildung 7).

5. **Zellen mit vier benachbarten leeren Zellen:**
   Hier werden alle Geschwindigkeiten an den Kanten der Zelle mit dem Flag `C_NSWO` nach (14) und (15) berechnet, und die 8 benachbarten Werte werden gleich den nächstliegenden, so bestimmten Werten gesetzt, sofern sie an leere Zellen angrenzen (z.B. $u_{i,j+1} = u_{i,j}$ oder $v_{i-1,j} = v_{i,j}$). Der Druck erhält wiederum den Wert Null.

6. **Zusammenfassung.**
   In Abildung 7 wird nochmals veranschaulicht, welche der in Abbildung 3 mit hellen Kreisen gekennzeichneten Geschwindigkeiten bei der Behandlung der jeweiligen Oberflächenzellen noch berechnet werden müssen, wenn die Zellen beim Setzen der Randwerte von links unten nach rechts oben durchlaufen werden.

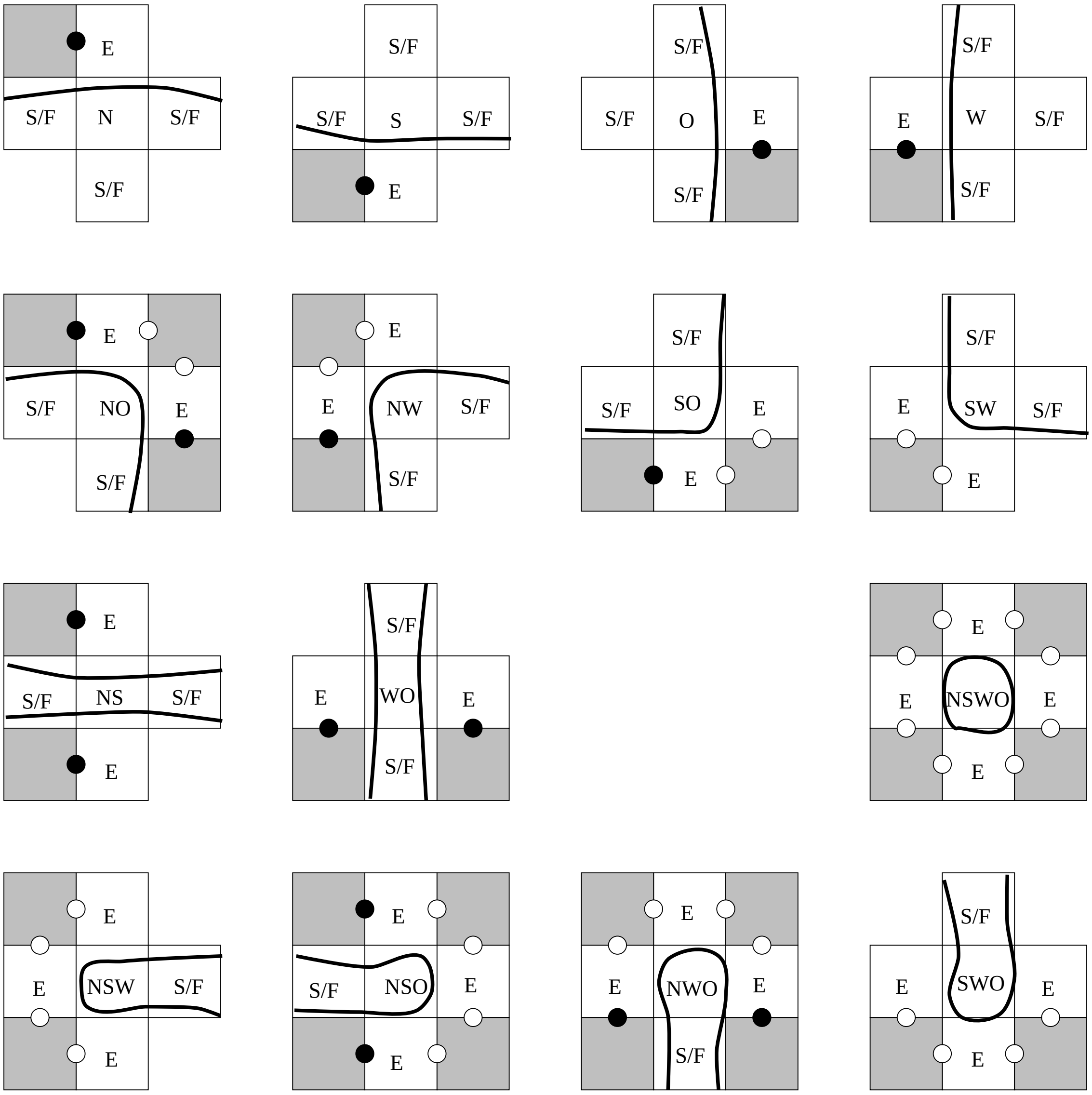


Abb. 7: Oberflächenzellen

Die Werte, die auf Grund der diskretisierten Version von Gleichung (2) berechnet werden, sind mit einem schwarzen Punkt markiert und die Werte, die analog zu (13) gesetzt werden, sind mit weißen Punkten markiert. Die entsprechenden Formeln werden aber immer nur dann angewendet, wenn die angrenzenden grau schattierten Zellen leer sind, da sonst die Geschwindigkeiten von diesen Zellen aus bestimmt werden. Das Kürzel "S/F" in Abbildung 7 soll jeweils eine Zelle kennzeichnen, die im Inneren des Flüssigkeitsgebietes oder am freien Rand liegt.

## 4. Der erweiterte Algorithmus

Wir wollen jetzt den für freie Ränder erweiterten Algorithmus zusammenfassen. Dabei sei noch erwähnt, daß vor dem Fortbewegen der Partikel die Werte am Rand, egal ob fest oder frei, nochmals gesetzt werden sollten, um alle Geschwindigkeiten zum Zeitpunkt $t_{n+1}$ zur Verfügung zu haben und nicht die Geschwindigkeiten im Inneren zum Zeitpunkt $t_{n+1}$ und am Rand zum Zeitpunkt $t_n$. Dabei genügt es, die Werte am festen Rand einmal vor der Zeitschleife als auch in der Zeitschleife *nach* Berechnung der neuen Geschwindigkeiten im Inneren zu berechnen, während die Randwerte am freien Rand zweimal gesetzt werden müssen: einmal nach der Berechnung der neuen Geschwindigkeiten aber auch nach der Bestimmung der neuen Gebietsform, da sich ja aufgrund der neuen Partikelpositionen das Gebiet ändern kann. Insgesamt erhalten wir:

Setze $t := 0$, $n := 0$
Setze die Partikel in das Anfangsgebiet $\Omega_0$
Belege $u, v, p$ mit Anfangswerten
Setze die Werte für $u$ und $v$ am festen Rand von $\Omega_0$
Solange $t < t_{end}$
    Wähle gegebenenfalls $\delta t$ gemäß (Blatt1,(19))
    Markiere die inneren und Oberflächenzellen von $\Omega_{t_n}$
    Bestimme die Werte für $u, v$ und $p$ am freien Rand von $\Omega_{t_n}$
    Berechne $F^{(n)}$ und $G^{(n)}$ gemäß (Blatt1,(9),(10),(17)) im Fluidgebiet
    Berechne die rechte Seite der Druckgleichung (Blatt1,(11)) für innere Zellen
    Setze $it := 0$
    Solange $it < itmax$ und $\|r^{it}\| > \varepsilon$
        Führe einen SOR-Zyklus gemäß (Blatt1,(12)) in den inneren Zellen durch
        Berechne die Norm des Residuums der Druckgleichung $\|r^{it}\|$
        $it := it + 1$
    Berechne $u^{(n+1)}$ und $v^{(n+1)}$ gemäß (Blatt1,(7),(8)) im Fluidgebiet
    Setze die Werte für $u^{(n+1)}$ und $v^{(n+1)}$ am festen Rand
    Bestimme die Werte für $u^{(n+1)}$ und $v^{(n+1)}$ am freien Rand von $\Omega_{t_n}$
    Berechne die Partikelpositionen zum Zeitpunkt $t_{n+1}$
    Gegebenenfalls Datenausgabe zur Visualisierung
    $t := t + \delta t$
    $n := n + 1$

## 5. Aufgabenstellung

1. Legen Sie eine Kopie der Prozeduren an, welche gemäß Blatt 1 und 2 erstellt wurden.

2. Erweitern Sie das Flagfeld geeignet. Dies kann zum Beispiel erfolgen, indem fünf weitere Bitpositionen hinzugefügt werden: eine für die Zelle selbst und vier für die Nachbarzellen. Diese werden auf Eins gesetzt, wenn die entsprechende Zelle leer ist oder auf Null, wenn die Zelle eine Fluid-, Oberflächen- oder Hinderniszelle ist. Insgesamt ergibt sich die Bitfolge

| 1=leer, 0=Fluid/Oberfl./Hind. | | | | | 1=Fluid/leer/Oberfl., 0=Hind. | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Mitte | Ost | West | Süd | Nord | Mitte | Ost | West | Süd | Nord |

3. Schreiben Sie folgende Prozeduren (z.B. in einer Datei `surface.c`):

   - `struct particleline *INIT_PARTICLES (int *N,int imax,int jmax,`
     `REAL delx,REAL dely,int ppc,char *problem,REAL **U,REAL **V)`
     Für die Probleme "Fallender Tropfen" und "Brechender Damm" sollen die Partikel der Ausgangskonfiguration wie in Abbildung 1 dargestellt gesetzt werden. Beim Tropfen soll dabei ein Zeiger auf ein Feld `Partlines` zurückgegeben werden, das aus 2 Listen besteht (N=2), wobei in der einen Liste die Partikel des Tropfens und in der anderen Liste die Partikel der Flüssigkeit im Becken enthalten sind. Die Höhe des Beckens beträgt $j_{\max} \cdot \delta y/3$, der Radius des Tropfens ist $j_{\max} \cdot \delta y/10$ und der Mittelpunkt des Tropfens soll bei $(i_{\max} \cdot \delta x/2, 2/3 \cdot j_{\max} \cdot dely)$ liegen. Die Fallgeschwindigkeit des Tropfens zum Startzeitpunkt betrage $u = 0$, $v = -2$.

     Beim Damm können dagegen alle Partikel in einer Liste gehalten werden (N=1). Die Breite des Dammes beträgt $i_{\max} \cdot \delta x/5$.

     Der Parameter `ppc` soll die Anzahl der potentiellen Partikelpositionen zum Startzeitpunkt in einer Zelle mit gleicher $y$–Koordinate angeben. Meist wird `ppc` den Wert 3 oder 4 annehmen, so daß am Anfang in einer Zelle, die ganz im Gebiet des Fluids liegt, 9 bzw. 16 Partikel gleichmäßig verteilt werden (vgl. Abschnitt 2).
     Der Aufruf erfolgt in `main` im Initialisierungsteil.

   - `void MARK_CELLS (int **FLAG,int imax,int jmax,REAL delx,REAL dely,`
     `int N,struct particleline *Partlines)`
     In einer *Schleife über alle Partikel* werden die Zellen, in denen mindestens ein Partikel liegt, mit dem Flag `C_F` versehen, während alle anderen Nicht-Hinderniszellen das Flag `C_E` erhalten.

     Anschließend werden in einer *Schleife über alle Zellen* die Typen der Oberflächenzellen bestimmt und mit den jeweiligen Flags versehen (siehe Abschnitt 2).
     Der Aufruf erfolgt zu Beginn der Zeitschleife von `main`.

   - `void SET_UVP_SURFACE (REAL **U,REAL **V,REAL **P,int **FLAG,int imax,`
     `int jmax,REAL Re,REAL delx,REAL dely,REAL delt)`

Die Randwerte für $u$, $v$ und $p$ werden, wie in Abschnitt 3 beschrieben, in Abhängigkeit von den jeweiligen Flags der Zellen berechnet. Dabei müssen zunächst alle Geschwindigkeitswerte bestimmt werden, bevor die Druckwerte berechnet werden.
Der Aufruf erfolgt in `main` in Anschluß an `MARK_CELLS`.

4. Führen Sie an den bestehenden Prozeduren folgende Modifikationen durch:

   - In `main`:
     Führen Sie den Parameter `int ppc` ein und fügen Sie den Aufruf der oben beschriebenen Prozeduren ein. Außerdem müssen am Ende der Zeitschleife die Partikel mit der Prozedur `ADVANCE_PARTICLES` weiterbewegt werden.
   - In `COMP_FG`:
     $F$ und $G$ werden nur für Kanten berechnet, die nicht an eine leere Zelle grenzen.
   - In `COMP_RHS`:
     Die rechte Seite der Druckgleichung wird nur für innere Zellen berechnet.
   - In `POISSON`:
     Bei der SOR-Iteration werden nur die Druckwerte in den inneren Zellen neu berechnet. Auch bei der Berechnung des Residuums dürfen nur die Anteile für die inneren Zellen addiert werden. Bei der Normierung ist dann nur durch die Anzahl der inneren Zellen zu teilen.
   - In `ADAP_UV`:
     Analog zu $F$ und $G$ werden die Geschwindigkeiten $u$ und $v$ zum neuen Zeitpunkt nur an Kanten berechnet, die nicht an leere oder Hinderniszellen angrenzen und somit noch nicht in `SET_UVP_SURFACE` oder `SET(SPEC)BCOND` neu bestimmt wurden.

5. Simulieren Sie den “Fallenden Tropfen” und den “Brechenden Damm”, indem Sie folgende Eingabeparameter verwenden:

   Für den Tropfen:

   | | | | |
   |---|---|---|---|
   | imax = 40 | jmax = 30 | xlength = 8.0 | ylength = 6.0 |
   | delt = 0.01 | t_end = 10.0 | tau = 0.2 | |
   | eps = 0.001 | omg = 1.7 | alpha = 0.5 | itermax = 500 |
   | GX = 0.0 | GY = -1.0 | Re = 40 | |
   | UI = 0.0 | VI = 0.0 | PI = 0.0 | ppc = 4 |
   | wl = 2 | wr = 2 | wb = 2 | wt = 3 |

   Für den Damm:

   | | | | |
   |---|---|---|---|
   | imax = 50 | jmax = 20 | xlength = 10.0 | ylength = 4.0 |
   | delt = 0.04 | t_end = 5.0 | tau = 0.5 | |
   | eps = 0.001 | omg = 1.7 | alpha = 0.5 | itermax = 500 |
   | GX = 0.0 | GY = -1.0 | Re = 10.0 | |
   | UI = 0.0 | VI = 0.0 | PI = 0.0 | ppc = 4 |
   | wl = 2 | wr = 3 | wb = 2 | wt = 3 |

   Zur Visualisierung kann das `MatLab`–Programm für die Streaklines verwendet werden, bei dem die Partikelpositionen wie gewünscht zu verschiedenen Zeitpunkten dargestellt werden, so daß ein kleiner Film abläuft. Die obige Technik hat den Nachteil, daß alle Partikel die gleiche Farbe haben und so nicht unterschieden werden kann, ob ein Partikel ursprünglich z.B. zum Tropfen oder zum Beckeninhalt gehört. Da wir aber in diesem Fall die Partikel in zwei unterschiedlichen Listen gehalten haben, wäre die Information 'Tropfen/Becken' eigentlich verfügbar. Die Idee ist nun, mittels `WRITE_PARTICLES` die beiden Listen *getrennt voneinander* in zwei unterschiedliche Dateien zu schreiben. Diese beiden Dateien werden dann mittels z.B. `p1=ReadParticles(<file_tropfen>);` `p2=ReadParticles(<file_becken>);` eingelesen. Zur Visualisierung kann man dann z.B.

   ```
   >>spy(p1',5.0,'g') ;hold on;
   >>spy(p2',5.0,'k') ;hold off;
   ```

   benutzen, siehe `>> help spy`.