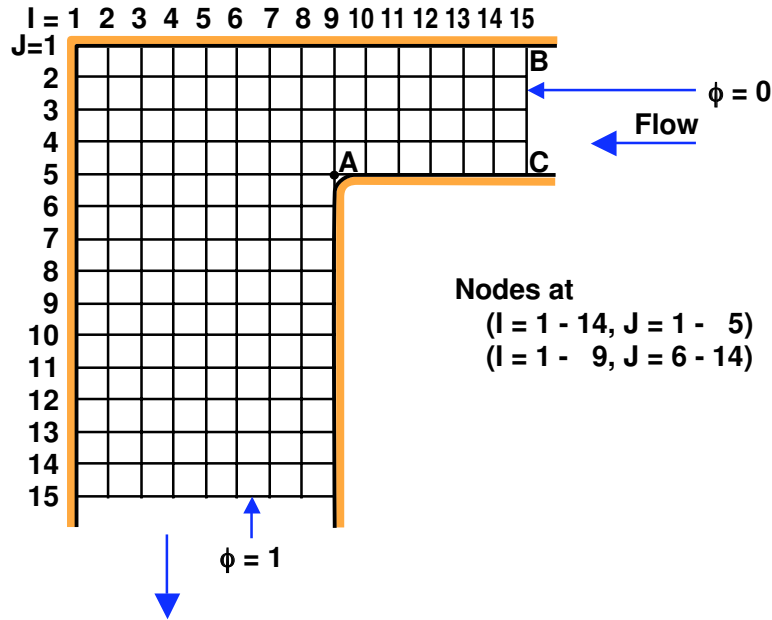


PROBLEM 30 (Value 4 times the regular homework problems.)

Obtain a numerical solution to the planar, potential flow around a bend using the grid sketched below.



The upstream and downstream boundaries to be used are $\phi = 0$ and $\phi = 1$ respectively. You are not required use any special procedures to treat the singular behaviour near the projecting corner. You are advised to use an iterative method with over-relaxation and, by trial and error, to find an effective value of the over-relaxation factor which improves the convergence of the iterative process. Find:

- (a) The velocity potential at all of the nodes and, in particular, the node *A*.
- (b) If the volume averaged velocity, U , at inlet (*BC*) is unity find the velocity distributions along both the interior and exterior walls. It is best to evaluate the velocity at points on the walls midway between the nodes.
- (c) If the pressure coefficient is defined as $(p - p_B)/\frac{1}{2}\rho U^2$ where p is the pressure, p_B is the pressure at the point *B* and ρ is the fluid density, find the pressure coefficient along both the interior and exterior walls. It is best to evaluate the pressures at the same points at which the velocities were evaluated.

SOLUTION 30

Obtain the numerical solution to the planar, potential flow around a bend.

- (a) Find the velocity potential at all of the nodes and, in particular, the node *A*.

Any neighboring point can be expanded in a Taylor's Series about a central point, $\phi_{i,j}$.

$$\phi_{i+1,j} = \phi_{i,j} + \frac{\partial \phi}{\partial x} \Big|_{i,j} l + \frac{1}{2} \frac{\partial^2 \phi}{\partial x^2} \Big|_{i,j} l^2 + \dots$$

$$\phi_{i-1,j} = \phi_{i,j} - \frac{\partial\phi}{\partial x}\Big|_{i,j} l + \frac{1}{2} \frac{\partial^2\phi}{\partial x^2}\Big|_{i,j} l^2 + \dots$$

By adding these equations, we can form an expression for $\frac{\partial^2\phi}{\partial x^2}$:

$$\frac{\partial^2\phi}{\partial x^2}\Big|_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} - 2\phi_{i,j}}{l^2}$$

Similarly the neighboring points in the y-direction can be expanded about the central point to give:

$$\frac{\partial^2\phi}{\partial y^2}\Big|_{i,j} = \frac{\phi_{i,j+1} + \phi_{i,j-1} - 2\phi_{i,j}}{l^2}$$

Using these two expressions to form Laplace's Equation, we get:

$$\begin{aligned} \nabla^2\phi &= \frac{\partial^2\phi}{\partial x^2}\Big|_{i,j} + \frac{\partial^2\phi}{\partial y^2}\Big|_{i,j} = \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}}{l^2} = 0 \\ \Rightarrow \phi_{i,j} &= \frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1}}{4} \end{aligned}$$

This expression holds at all points in the interior of the grid.

Along a wall, we have a slightly different relationship. At such a point, the velocity normal to the wall is zero. We can no longer write an expansion for $\phi_{i-1,j}$ and we know:

$$u = \frac{\partial\phi}{\partial x}\Big|_{i,j} = 0$$

So the expansion for $\phi_{i+1,j}$ becomes:

$$\begin{aligned} \phi_{i+1,j} &= \phi_{i,j} + \frac{1}{2} \frac{\partial^2\phi}{\partial x^2}\Big|_{i,j} l^2 + \dots \\ \Rightarrow \frac{\partial^2\phi}{\partial x^2}\Big|_{i,j} &= \frac{2\phi_{i+1,j} - 2\phi_{i,j}}{l^2} \end{aligned}$$

Again forming Laplace's equation, we see that along such a wall we have:

$$\phi_{i,j} = \frac{2\phi_{i+1,j} + \phi_{i,j+1} + \phi_{i,j-1}}{4}$$

(b) Find the velocity distributions along both the interior and exterior walls.

The velocity midway between two nodes is the difference in ϕ over the cell spacing, l .

$$\begin{aligned} u_{i,j} &= \frac{\partial\phi}{\partial x}\Big|_{i,j} = \frac{\phi_{i,j} - \phi_{i+1,j}}{l} \\ v_{i,j} &= \frac{\partial\phi}{\partial y}\Big|_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j}}{l} \end{aligned}$$

To find the volume averaged velocity, we sum the flowrates across the inlet and divide by the total area. Each flowrate is given by the velocity midway between two nodes (vertically) multiplied by the cell spacing, l .

$$U = \sum_{j=1}^5 \frac{q_j}{A} = \frac{\frac{1}{2}\phi_{14,1} + \phi_{14,2} + \phi_{14,3} + \phi_{14,4} + \frac{1}{2}\phi_{14,5}}{4l} = \frac{\bar{Q}}{l}$$

Scaling all of the velocities by the volume averaged velocity:

$$\hat{u}_{i,j} = \frac{u_{i,j}}{U} = \frac{\phi_{i,j} - \phi_{i+1,j}}{\bar{Q}}$$

$$\hat{v}_{i,j} = \frac{v_{i,j}}{U} = \frac{\phi_{i,j+1} - \phi_{i,j}}{\bar{Q}}$$

(c) Find the pressure coefficient, $(p - p_B)/\frac{1}{2}\rho U^2$, along both the interior and exterior walls.

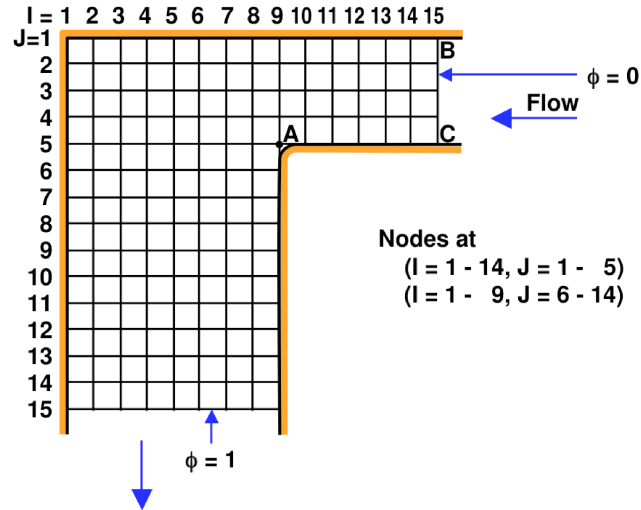
Apply Bernoulli's equation at point B and any other point in the flow.

$$\begin{aligned} \frac{1}{2}\rho|\vec{u}_B|^2 + p_B &= \frac{1}{2}\rho|\vec{u}|^2 + p \\ \Rightarrow \frac{(p - p_B)}{\frac{1}{2}\rho U^2} &= \frac{V_B^2}{U^2} - \frac{V^2}{U^2} = \hat{u}_{14,1}^2 - (\hat{u}_{i,j}^2 + \hat{v}_{i,j}^2) \end{aligned}$$

This notebook is used to solve the potential flow problem given as me19a-hw8:

PROBLEM 30 (Value 4 times the regular homework problems.)

Obtain a numerical solution to the planar, potential flow around a bend using the grid sketched below.



The upstream and downstream boundaries to be used are $\phi = 0$ and $\phi = 1$ respectively. You are not required use any special procedures to treat the singular behaviour near the projecting corner. You are advised to use an iterative method with over-relaxation and, by trial and error, to find an effective value of the over-relaxation factor which improves the convergence of the iterative process. Find:

- The velocity potential at all of the nodes and, in particular, the node A.
- If the volume averaged velocity, U , at inlet (BC) is unity find the velocity distributions along both the interior and exterior walls. It is best to evaluate the velocity at points on the walls midway between the nodes.
- If the pressure coefficient is defined as $(p - p_B)/0.5 \rho U^2$ where p is the pressure, p_B is the pressure at the point B and ρ is the fluid density, find the pressure coefficient along both the interior and exterior walls. It is best to evaluate the pressures at the same points at which the velocities were evaluated.

The solution method is outlined in the file *me19a-hw8-solutions.pdf* and is implemented below.

- (a) Find the velocity potential at all of the nodes and, in particular, the node A.

A sensitivity analysis will reveal that a relaxation factor of $\alpha = 1.825$ will lead to fastest convergence, without regard for the initialized value of the computational domain. This value is used below and the computational domain is initialized to 0.6.

```

In[1]:= (* initialize domain *)
phi = Table[0.6, {15}, {15}];

(* set inlet condition at nodes, i=15,j=1-5 *)
phi[[15, 1 ;; 5]] = 0;

(* set outlet condition at nodes, i=1-9,j=15 *)
phi[[1 ;; 9, 15]] = 1;

(* relaxation value *)
alpha = 1.825;

(* counter to track number of iterations *)
count = 0;

(* allowable error to check convergence *)
tolerance = 10^-6;

(* iteration loop *)
phiNew = phi; While[True, {

  totErrorSq = 0;

  For[j = 14, j > 0, j--,
    For[i = 14, i > 0, i--, {

      (* solve only for i,j in domain of interest *)
      If[!(i > 9 && j > 5), {

        If[i == 1 && j == 1 (* top left corner *),
          phiNew[[i, j]] = 0.25 * (2 * phi[[i + 1, j]] + 2 * phi[[i, j + 1]]),

        (* else1 *) If[j == 1 (* top wall *),
          phiNew[[i, j]] = 0.25 * (phi[[i + 1, j]] + phi[[i - 1, j]] + 2 * phi[[i, j + 1]]),

        (* else2 *) If[i == 1 (* left wall *),
          phiNew[[i, j]] = 0.25 * (phi[[i, j + 1]] + phi[[i, j - 1]] + 2 * phi[[i + 1, j]]),

        (* else3 *) If[i == 9 && j > 5 (* right wall *),
          phiNew[[i, j]] = 0.25 * (phi[[i, j + 1]] + phi[[i, j - 1]] + 2 * phi[[i - 1, j]]),

        (* else4 *) If[i > 9 && j == 5 (* bottom wall *),
          phiNew[[i, j]] = 0.25 * (phi[[i + 1, j]] + phi[[i - 1, j]] + 2 * phi[[i, j - 1]]),

        (* else the rest are interior nodes *)

```

```

phiNew[[i, j]] =
  0.25 * (phi[[i + 1, j]] + phi[[i - 1, j]] + phi[[i, j + 1]] + phi[[i, j - 1]])

  ](* end4 *)
  ](* end3 *)
  ](* end2 *)
  ](* end1 *)
](* end *);

(* use relaxation constant to calculate new phi *)
phiNew[[i, j]] = (1 - alpha) * phi[[i, j]] + alpha * phiNew[[i, j]];

(* sum the errors *)
errorSq = (phiNew[[i, j]] - phi[[i, j]])^2;
totErrorSq += errorSq;

(* update solution for phi *)
phi[[i, j]] = phiNew[[i, j]];

  ](* end if *);

  ](* end for i *)
](* end for j *);

count++;

(* convergence check *)
If[Sqrt[totErrorSq] < tolerance, Break[]];

];(* end while *)

Print["Domain initialized to ", phi[[15, 15]]]
Print["Solution converged in ", count, " iterations with relaxation constant ", alpha]
Print["Total error is ", Sqrt[totErrorSq]]
Print["The solution at node A is ", phi[[9, 5]]]

```

Domain initialized to 0.6

Solution converged in 74 iterations with relaxation constant 1.825

Total error is 9.65797×10^{-7}

The solution at node A is 0.499376

```

In[12]:= solution = Table[phi[[i, j]], {j, 15}, {i, 15}];
Style[TableForm[solution /. {phi[[15, 15]] -> ""},
  TableHeadings -> {Table[j, {j, 15}], Table[i, {i, 15}]}], 6.5]

```

```

Out[13]=


|    | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        | 10       | 11       | 12       | 13       | 14        | 15 |
|----|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|----|
| 1  | 0.631358 | 0.628469 | 0.619706 | 0.604778 | 0.5832   | 0.554314 | 0.517371 | 0.471753 | 0.417409 | 0.355441 | 0.288223 | 0.217904 | 0.145922 | 0.0731251 | 0  |
| 2  | 0.634247 | 0.631407 | 0.622789 | 0.608103 | 0.586854 | 0.558342 | 0.521709 | 0.476116 | 0.42122  | 0.358067 | 0.289774 | 0.218734 | 0.14633  | 0.0732893 | 0  |
| 3  | 0.642818 | 0.640121 | 0.631939 | 0.617991 | 0.597771 | 0.570491 | 0.535005 | 0.489783 | 0.433289 | 0.365831 | 0.294073 | 0.220929 | 0.147374 | 0.073702  | 0  |
| 4  | 0.656783 | 0.65432  | 0.646856 | 0.63415  | 0.615747 | 0.590848 | 0.558039 | 0.514722 | 0.456321 | 0.377896 | 0.299758 | 0.223535 | 0.148534 | 0.0741448 | 0  |
| 5  | 0.675674 | 0.673521 | 0.667014 | 0.656008 | 0.640219 | 0.619114 | 0.591579 | 0.554745 | 0.499376 | 0.389674 | 0.303528 | 0.22492  | 0.149083 | 0.074343  | 0  |
| 6  | 0.698873 | 0.697074 | 0.691671 | 0.682649 | 0.670006 | 0.653811 | 0.634419 | 0.613305 | 0.596763 |          |          |          |          |           |    |
| 7  | 0.72567  | 0.724232 | 0.719946 | 0.712911 | 0.703344 | 0.691706 | 0.678981 | 0.667292 | 0.661067 |          |          |          |          |           |    |
| 8  | 0.755343 | 0.754238 | 0.750972 | 0.745704 | 0.738752 | 0.73069  | 0.722505 | 0.715815 | 0.71292  |          |          |          |          |           |    |
| 9  | 0.787225 | 0.786405 | 0.783999 | 0.78018  | 0.775272 | 0.769797 | 0.764536 | 0.760543 | 0.758982 |          |          |          |          |           |    |
| 10 | 0.820748 | 0.820158 | 0.818438 | 0.815746 | 0.812358 | 0.80869  | 0.805298 | 0.802839 | 0.801923 |          |          |          |          |           |    |
| 11 | 0.855451 | 0.855041 | 0.85385  | 0.852007 | 0.849724 | 0.847306 | 0.845127 | 0.843591 | 0.843033 |          |          |          |          |           |    |
| 12 | 0.890975 | 0.890703 | 0.889916 | 0.888707 | 0.887227 | 0.885682 | 0.884315 | 0.883366 | 0.883026 |          |          |          |          |           |    |
| 13 | 0.927045 | 0.92688  | 0.926404 | 0.925677 | 0.924794 | 0.923882 | 0.923082 | 0.922534 | 0.922338 |          |          |          |          |           |    |
| 14 | 0.963445 | 0.963367 | 0.963143 | 0.962803 | 0.962391 | 0.961968 | 0.9616   | 0.961348 | 0.961258 |          |          |          |          |           |    |
| 15 | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        | 1        |          |          |          |          |           |    |


```

```
In[14]:= phi0 = Table[0.6, {15}, {15}];  
phi0[[14, 1 ;; 5]] = 0;  
phi0[[1 ;; 9, 14]] = 1;  
pphi0 = Show[  
  ListContourPlot[Flatten[Table[{i, -j, phi0[[i, j]]}, {j, 14}, {i, 14}], 1],  
  Contours -> 20, ContourStyle -> None, ColorFunction -> "Rainbow"],  
  Graphics[{White, Rectangle[{9, -15}, {15, -5}]}],  
  FrameTicks -> None, Frame -> False, AspectRatio -> 1  
];  
  
In[18]:= pphi = Show[  
  ListContourPlot[Flatten[Table[{i, -j, phi[[i, j]]}, {j, 14}, {i, 14}], 1],  
  Contours -> 50, ColorFunction -> "Rainbow"],  
  Graphics[{White, Rectangle[{9, -15}, {15, -5}]}],  
  FrameTicks -> None, Frame -> False, AspectRatio -> 1  
];
```

- (b) Find the velocity distributions along both the interior and exterior walls.

Since velocities are calculated between nodes, $u[[i, j]]$ corresponds to the velocity between nodes i and $i + 1$, and $v[[i, j]]$ corresponds to the velocity between nodes $j + 1$ and j .

```
In[19]:= dx = dy = 1;

(* initialize u *)
u = Table[0, {14}, {14}];

(* initialize v *)
v = Table[0, {14}, {14}];

For[j = 14, j > 0, j--,
  For[i = 14, i > 0, i--, {
    (* solve only for i,j in domain of interest *)
    If[!(i > 9 && j > 5), {
      u[[i, j]] = (phi[[i, j]] - phi[[i + 1, j]]) / dx;
      v[[i, j]] = (phi[[i, j + 1]] - phi[[i, j]]) / dy
    }>(* end if *)
  }>(* end for i *)
>(* end for j *)];

UInlet = 0.25 * (0.5 * u[[14, 1]] + u[[14, 2]] + u[[14, 3]] + u[[14, 4]] + 0.5 * u[[14, 5]]);

u = u / UInlet;
u[[9, 6 ;; 14]] = 0;

v = v / UInlet;
v[[10 ;; 14, 5]] = 0;
```

```
In[28]:= usolution = Table[u[[i, j]], {j, 14}, {i, 14}];
Style[TableForm[usolution /. {0 -> ""}, TableHeadings ->
  {Table[j, {j, 15}], Table[ToString[i] <> "-" <> ToString[i + 1], {i, 14}]}], 6.5]
```

```
Out[29]=
```

	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15
1	0.0391903	0.118877	0.202507	0.292713	0.391849	0.501139	0.61882	0.737197	0.840604	0.911831	0.95391	0.97645	0.987513	0.991964
2	0.0385363	0.116906	0.199219	0.288249	0.386773	0.496941	0.618474	0.744681	0.856693	0.926406	0.96368	0.982187	0.990818	0.99419
3	0.0365875	0.110991	0.189214	0.27429	0.370051	0.481378	0.613454	0.766362	0.915082	0.973417	0.992219	0.997802	0.999381	0.999789
4	0.0334119	0.101256	0.172354	0.249648	0.337763	0.445066	0.587603	0.792232	1.06385	1.05996	1.03398	1.01741	1.00911	1.0058
5	0.0292168	0.0882678	0.149298	0.214186	0.286286	0.37352	0.499663	0.751107	1.48814	1.1686	1.06633	1.02876	1.01387	1.00848
6	0.0244043	0.073299	0.122386	0.171508	0.21968	0.263062	0.286421	0.224395						
7	0.0195059	0.0581385	0.0954407	0.129777	0.157863	0.172629	0.158564	0.0844473						
8	0.0149884	0.0443072	0.0714607	0.0942963	0.109365	0.11103	0.0907594	0.0392753						
9	0.0111271	0.0326408	0.0518021	0.0665801	0.0742691	0.0713676	0.0541652	0.0211716						
10	0.0080059	0.023329	0.0365237	0.0459567	0.049761	0.0460101	0.0333574	0.0124207						
11	0.0055701	0.0161458	0.0250088	0.0309621	0.0328072	0.0295501	0.0208375	0.00757453						
12	0.00369747	0.0106746	0.0164059	0.0200734	0.0209529	0.0185523	0.0128633	0.0046192						
13	0.0022411	0.00645363	0.00986306	0.0119745	0.0123764	0.0108439	0.00744498	0.00265449						
14	0.00105417	0.00303336	0.0046191	0.00558377	0.00573956	0.00499991	0.00341475	0.00121303						

Note that $u[[i, 15]] = 0$ for all i because $\text{phi}[[i, 15]] = \text{phi}[[i + 1, 15]] = 1$ at the outlet. Positive u is defined to be to the left.


```
In[30]:= vsolution = Table[v[[i, j]], {j, 14}, {i, 14}];
Style[TableForm[vsolution /. {0 -> ""}, TableHeadings ->
{Table[ToString[j] <> "-" <> ToString[j + 1], {j, 14}], Table[i, {i, 14}]}], 6.5]
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1-2	0.0391897	0.0398437	0.0418146	0.0451029	0.0495674	0.0546437	0.0588417	0.0591877	0.0517035	0.0356142	0.0210395	0.011269	0.00553158	0.002226
2-3	0.116264	0.118213	0.124128	0.134133	0.148091	0.164813	0.180376	0.185396	0.163715	0.105326	0.0583143	0.0297757	0.0141614	0.005598
3-4	0.18944	0.192616	0.202351	0.21921	0.243852	0.27614	0.312452	0.338303	0.312434	0.163661	0.0771156	0.0353534	0.015741	0.006007
4-5	0.256265	0.26046	0.273448	0.296505	0.331967	0.383444	0.45499	0.542931	0.584055	0.15977	0.051135	0.0187868	0.00744329	0.002688
5-6	0.314698	0.31951	0.334479	0.361391	0.404069	0.470675	0.581133	0.794375	1.32109					
6-7	0.363507	0.368405	0.383565	0.410511	0.452242	0.514059	0.604492	0.732349	0.872297					
7-8	0.402519	0.407037	0.420868	0.444848	0.480328	0.528826	0.590426	0.65823	0.703402					
8-9	0.432494	0.436355	0.448021	0.46768	0.495396	0.530492	0.570154	0.606748	0.624852					
9-10	0.45475	0.457871	0.467183	0.482461	0.503084	0.527592	0.55295	0.573758	0.582509					
10-11	0.470759	0.473194	0.480378	0.491892	0.506887	0.523841	0.540301	0.552821	0.557667					
11-12	0.481898	0.48377	0.489241	0.497844	0.508733	0.520587	0.531585	0.539559	0.542515					
12-13	0.489293	0.49075	0.494971	0.501513	0.509612	0.518189	0.525897	0.531315	0.53328					
13-14	0.493774	0.494961	0.498381	0.503625	0.510016	0.516653	0.522497	0.526527	0.527968					
14-15	0.495883	0.496937	0.49997	0.504589	0.510173	0.515913	0.520913	0.524327	0.52554					

Note that $v[[15, j]] = 0$ for all j because $\text{phi}[[15, j]] = \text{phi}[[15, j + 1]] = 0$ at the inlet. Positive v direction is down.

```
In[32]:= pu = Show[
  ListContourPlot[Flatten[Table[{i, -j, u[[i, j]]}, {j, 14}, {i, 14}], 1],
  Contours -> 100, ContourStyle -> None, ColorFunction -> "Rainbow",
  Graphics[{White, Rectangle[{9, -14}, {14, -5}]}],
  FrameTicks -> None, Frame -> False, AspectRatio -> 1
];
```

```
In[33]:= pv = Show[
  ListContourPlot[Flatten[Table[{i, -j, v[[i, j]]}, {j, 14}, {i, 14}], 1],
  Contours -> 100, ContourStyle -> None, ColorFunction -> "Rainbow",
  Graphics[{White, Rectangle[{9, -14}, {14, -5}]}],
  FrameTicks -> None, Frame -> False, AspectRatio -> 1
];
```

```
In[34]:= pvel = Show[
  ListContourPlot[Flatten[Table[{i, -j, Sqrt[u[[i, j]]^2 + v[[i, j]]^2}], {j, 14}, {i, 14}],
  1], Contours -> 100, ContourStyle -> None, ColorFunction -> "Rainbow",
  Graphics[{White, Rectangle[{9, -14}, {14, -5}]}],
  FrameTicks -> None, Frame -> False, AspectRatio -> 1
];
```

- (c) Find the pressure coefficient, $(p - p_B)/0.5 \rho U^2$, along both the interior and exterior walls.

Along a streamline from B we use Bernoulli's to obtain $p_B + 1/2 \rho u_B^2 = p + 1/2 \rho (u^2 + v^2)$ or rearranging $c_p = (p - p_B)/0.5 \rho U^2 = (u_B^2 - (u^2 + v^2))/U^2$. Note that we have already normalized our calculated velocities, so $cp[[i, j]] = u[[14, 1]]^2 - (u[[i, j]]^2 + v[[i, j]]^2)$.

```
In[35]:= (* initialize cp *)
cp = Table[0, {14}, {14}];

For[j = 14, j > 0, j--,
  For[i = 14, i > 0, i--, {

    (* solve only for i,j in domain of interest *)
    If[!(i > 9 && j > 5), {

      cp[[i, j]] = (u[[14, 1]]^2 + v[[14, 1]]^2) - (u[[i, j]]^2 + v[[i, j]]^2)

    } (* end if *)

  } (* end for i *)
] (* end for j *)

In[37]:= cpsolution = Table[cp[[i, j]], {j, 14}, {i, 14}];
Style[TableForm[cpsolution /. {0 -> ""},
  TableHeadings -> {Table[ToString[j] <> "-" <> ToString[j + 1], {j, 14}],
    Table[ToString[i] <> "-" <> ToString[i + 1], {i, 14}]}], 6.5]

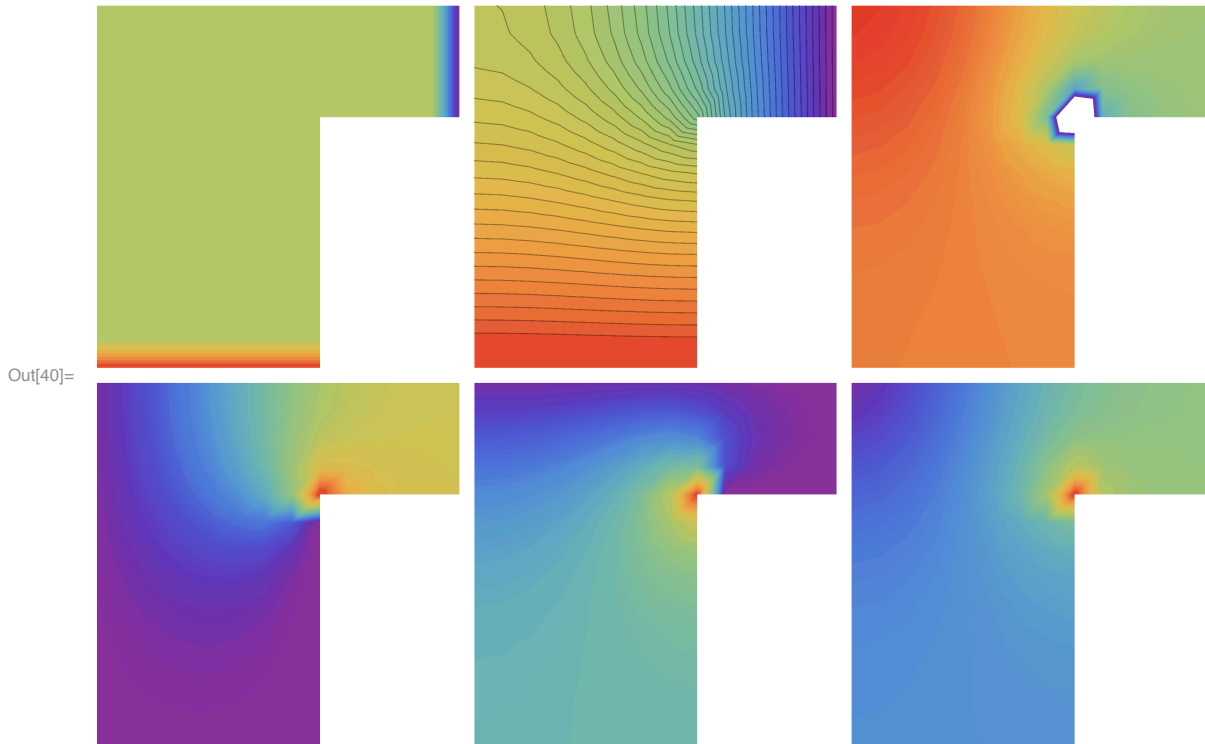
Out[38]=
```

	1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10	10-11	11-12	12-13	13-14	14-15
1-2	0.980925	0.968277	0.941239	0.896281	0.827994	0.729871	0.597596	0.437034	0.274709	0.151293	0.0736102	0.0304155	0.00878346	0.
2-3	0.968994	0.956355	0.928901	0.882918	0.812472	0.709883	0.568951	0.395075	0.223271	0.114676	0.0519165	0.0184183	0.00207555	-0.00444913
3-4	0.94677	0.934577	0.907249	0.860708	0.787595	0.676018	0.510044	0.282236	0.0490059	0.00967121	-0.00644849	-0.0128611	-0.0150125	-0.0156181
4-5	0.917208	0.905904	0.879517	0.833757	0.75971	0.638884	0.431703	0.0615914	-0.488911	-0.16505	-0.087735	-0.0514874	-0.0343708	-0.0276369
5-6	0.884108	0.874119	0.84983	0.807518	0.738765	0.622944	0.396618	-0.211197	-2.97584	-0.381624	-0.153061	-0.0743453	-0.0439352	-0.0330441
6-7	0.851264	0.842902	0.821896	0.786063	0.731215	0.650539	0.536549	0.397308	0.223095					
7-8	0.821595	0.814938	0.797758	0.769265	0.728361	0.674538	0.610252	0.543599	0.489222					
8-9	0.796721	0.791628	0.778167	0.75638	0.726619	0.690248	0.650684	0.614311	0.593557					
9-10	0.777076	0.773286	0.763054	0.746795	0.725387	0.70055	0.675309	0.654351	0.64468					
10-11	0.762319	0.75954	0.7519	0.739926	0.724586	0.70747	0.690959	0.678231	0.673004					
11-12	0.75174	0.749702	0.744014	0.735189	0.724111	0.712112	0.70098	0.692815	0.689674					
12-13	0.744575	0.743048	0.738732	0.732078	0.723853	0.715133	0.707263	0.701679	0.699609					
13-14	0.740179	0.738969	0.735516	0.730215	0.723727	0.716949	0.710939	0.706759	0.705246					
14-15	0.738096	0.737041	0.734005	0.729355	0.723687	0.717806	0.712635	0.709076	0.707804					

```
In[39]:= pcp = Show[
  ListContourPlot[Flatten[Table[{i, -j, cp[[i, j]]}, {j, 14}, {i, 14}], 1],
    Contours -> 100, ContourStyle -> None, ColorFunction -> "Rainbow",
    Graphics[{White, Rectangle[{9, -14}, {14, -5}]}],
    FrameTicks -> None, Frame -> False, AspectRatio -> 1
];
```

■ Colorful contour plots

```
In[40]:= GraphicsGrid[{{pphi0, pphi, pcp}, {pu, pv, pvel}}, Spacings -> 0]
```



The top row shows (left) the initialized computational domain with the given boundary conditions, (center) the solution for the velocity potential where spacing of the contour lines indicates the relative flow velocity, and (right) the pressure coefficient --- there is a large pressure drop around the corner at node A. The bottom row of images shows (left) the u velocity component, (center) the v velocity component, and (right) the speed or magnitude of the velocity.

```

% assign boundary conditions
outlet_ic = 1;
inlet_ic = 0;

count = 0;
tolerence = 1e-4;
alpha = 1.825;
TotErrSq = 0;
converge = 0;

% set initial conditions
for i=1:15
    for j=1:15
        phi(i,j) = 0;
    end
end
for i=1:9
    phi(i,15) = outlet_ic;
end

phi2 = phi;

while(~converge)

    TotErrSq = 0;

    for j=14:-1:1
        for i=14:-1:1

            % make sure the point is in the domain of interest
            if( ~(j>5) && (i>9)) )

                % top left corner
                if((i==1) && (j==1))
                    phi2(1,j) = 0.25*(2*phi(i+1,j)+2*phi(i,j+1));

                % top wall
                elseif(j==1)
                    phi2(i,j) = 0.25*( phi(i+1,j)+phi(i-1,j)+2*phi(i,j+1) );

                % left wall
                elseif(i==1)
                    phi2(i,j) = 0.25*(phi(i,j+1)+phi(i,j-1)+2*phi(i+1,j));

                % right wall
                elseif((i==9) && (j>5))
                    phi2(i,j)= 0.25*(phi(i,j+1)+phi(i,j-1)+2*phi(i-1,j));

                % bottom wall (right)

```

```

elseif((j==5) && (i>9))
    phi2(i,j) = 0.25*(phi(i+1,j)+phi(i-1,j)+2*phi(i,j-1) );

% all other points
else
    phi2(i,j) = 0.25*(phi(i-1,j) + phi(i+1,j) + phi(i,j-1) + phi(i,j+1));

end

% calculate new phi
phi2(i,j)=(1-alpha)*phi(i,j) + alpha*phi2(i,j);

% sum the errors
err = (phi2(i,j) - phi(i,j))^2;
TotErrSq = TotErrSq + err;

phi(i,j) = phi2(i,j);
end

end

end

count = count + 1;

% check for convergence
if(TotErrSq < tolerance)
    converge = 1;
end

end

count
TotErrSq
contourf(phi);
colorbar;

% Calculate the volume averaged velocity at inlet

i=14;
for j=1:5
    ui(j) = phi(i,j) - phi(i+1,j);
end
U_inlet = (0.5*ui(1)+ui(2)+ui(3)+ui(4)+0.5*ui(5))/4;

% Calculate velocity (divided by U_inlet) at point B
uB = (phi(14,1) - phi(15,1)) / U_inlet;

% calculate the u (x-direction) component of velocity between nodes (center
% of cell) along the walls

```

```

% top wall
j=1;
for i=1:14
    u(i,j) = (phi(i,j) - phi(i+1,j)) / U_inlet;
    cp(i,j) = uB^2 - u(i,j)^2;
end

% bottom, right wall
j=5;
for i=10:14
    u(i,j) = (phi(i,j) - phi(i+1,j)) / U_inlet;
    cp(i,j) = uB^2 - u(i,j)^2;
end

% left wall
i=1;
for j=1:14
    v(i,j) = (phi(i,j+1) - phi(i,j)) / U_inlet;
    cp(i,j) = uB^2 - v(i,j)^2;
end

% right wall
i=9;
for j=6:14
    v(i,j) = (phi(i,j+1) - phi(i,j)) / U_inlet;
    cp(i,j) = uB^2 - v(i,j)^2;
end

% print all phi values
for j=1:15
    phi(:,j)
end

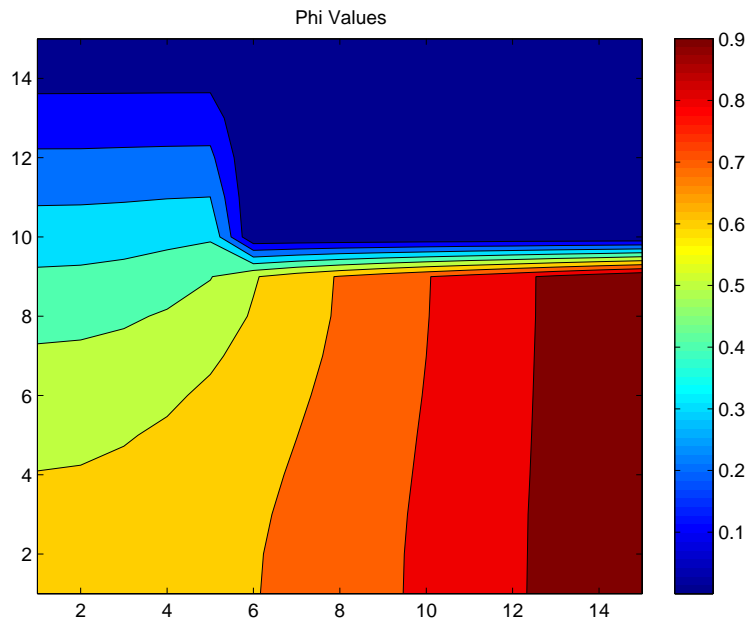
% print top wall
u(:,1)
cp(:,1)

%print bottom wall
u(10:14,5)
cp(10:14,5)

%print left wall
v(1,:)
cp(1,:)

%print right wall
v(9,6:14)
cp(9,6:14)

```



```
alpha = 1.83
count =
    77
TotErrSq =
    5.9308e-13
```

```
%% Phi values by row
ans =
```

```
0.6314
0.6285
0.6197
0.6048
0.5832
0.5543
0.5174
0.4718
0.4174
0.3554
0.2882
0.2179
0.1459
0.0731
0
```

```
ans =
```

0.6342
0.6314
0.6228
0.6081
0.5869
0.5583
0.5217
0.4761
0.4212
0.3581
0.2898
0.2187
0.1463
0.0733
0

ans =

0.6428
0.6401
0.6319
0.6180
0.5978
0.5705
0.5350
0.4898
0.4333
0.3658
0.2941
0.2209
0.1474
0.0737
0

ans =

0.6568
0.6543
0.6469
0.6342
0.6157
0.5908
0.5580
0.5147
0.4563
0.3779
0.2998
0.2235
0.1485

0.0741
0

ans =

0.6757
0.6735
0.6670
0.6560
0.6402
0.6191
0.5916
0.5547
0.4994
0.3897
0.3035
0.2249
0.1491
0.0743
0

ans =

0.6989
0.6971
0.6917
0.6826
0.6700
0.6538
0.6344
0.6133
0.5968
0
0
0
0
0
0

ans =

0.7257
0.7242
0.7199
0.7129
0.7033
0.6917
0.6790
0.6673

0.6611
0
0
0
0
0
0

ans =

0.7553
0.7542
0.7510
0.7457
0.7388
0.7307
0.7225
0.7158
0.7129
0
0
0
0
0
0

ans =

0.7872
0.7864
0.7840
0.7802
0.7753
0.7698
0.7645
0.7605
0.7590
0
0
0
0
0
0

ans =

0.8207
0.8202
0.8184

0.8157
0.8124
0.8087
0.8053
0.8028
0.8019
0
0
0
0
0
0

ans =

0.8555
0.8550
0.8539
0.8520
0.8497
0.8473
0.8451
0.8436
0.8430
0
0
0
0
0
0

ans =

0.8910
0.8907
0.8899
0.8887
0.8872
0.8857
0.8843
0.8834
0.8830
0
0
0
0
0
0

ans =

0.9270
0.9269
0.9264
0.9257
0.9248
0.9239
0.9231
0.9225
0.9223
0
0
0
0
0
0

ans =

0.9634
0.9634
0.9631
0.9628
0.9624
0.9620
0.9616
0.9613
0.9613
0
0
0
0
0
0

ans =

1
1
1
1
1
1
1
1
1
1
0
0
0

```
0
0
0
```

```
%% top wall velocity (between nodes)
```

```
ans =
```

```
0.0392
0.1189
0.2025
0.2927
0.3918
0.5011
0.6188
0.7372
0.8406
0.9118
0.9539
0.9764
0.9875
0.9920
```

```
%% top wall Cp
```

```
ans =
```

```
0.9825
0.9699
0.9430
0.8983
0.8304
0.7329
0.6011
0.4405
0.2774
0.1526
0.0740
0.0305
0.0088
```

```
0 (this is point B)
```

```
%% bottom wall u
```

```
ans =
```

```
1.1686
1.0663
1.0288
1.0139
1.0085
```

```
%% bottom wall Cp
```

```
ans =
```

-0.3816
-0.1531
-0.0743
-0.0439
-0.0331

%% left wall v
ans =

Columns 1 through 9

0.0392 0.1163 0.1894 0.2563 0.3147 0.3635 0.4025 0.4325 0.4547

Columns 10 through 14

0.4708 0.4819 0.4893 0.4938 0.4959

%% left wall Cp
ans =

Columns 1 through 9

0.9825 0.9705 0.9481 0.9183 0.8850 0.8519 0.8220 0.7969 0.7772

Columns 10 through 14

0.7624 0.7518 0.7446 0.7402 0.7381

%% right wall v
ans =

0.8723 0.7034 0.6249 0.5825 0.5577 0.5425 0.5333 0.5280 0.5255

%% right wall Cp
ans =

0.2231 0.4892 0.5936 0.6447 0.6730 0.6897 0.6996 0.7052 0.7078