Solving Incompressible Navier–Stokes in wxMaxima: A Day-by-Day Guide

This tutorial shows how to set up and analyze the incompressible 2D/3D Navier–Stokes equations symbolically and numerically in wxMaxima. We use parametric coordinates x(u,v,w,t),y(u,v,w,t),z(u,v,w,t) and demonstrate defining velocity/pressure fields, substituting divergence free test functions, treating the Reynolds number Re(t) symbolically, evaluating expressions at sample points, verifying the incompressibility condition, and visualizing results (vector and scalar fields). All code is given in wxMaxima syntax. References are cited for the mathematical background and Maxima usage.

Day 1: Formulate the Incompressible Navier–Stokes Equations

- **Define symbols and variables.** Incompressible flow satisfies the continuity equation ∇⋅u=0en.wikipedia.org. Let u=(ux,uy,uz) be the velocity and p the pressure. For constant density ρ and viscosity μ, the incompressible Navier–Stokes momentum equations in vector form are

  ρ(∂tu+(u⋅∇)u)=−∇p+μ∇2u+ρg,

  and ∇⋅u=0en.wikipedia.orghplgit.github.io. In components (2D case), for (ux,uy):

  ρ(∂tux+ux∂xux+uy∂yux)=−∂xp+μ(∂xxux+∂yyux)+ρgx,

  and similarly for the y-component (seeen.wikipedia.org for full form).

- **Maxima setup.** In wxMaxima, declare u(x,y,t), v(x,y,t), p(x,y,t) as functions and parameters (set ρ, μ as symbols). For example:

  ```
  maxima
  CopyEdit
  (%i1) depends([u,v,p],[x,y,t]);  /* declare u,v,p dependent on x,y,t */
  (%i2) rho: 'rho; mu: 'mu;         /* constant density, viscosity */
  ```

  Then define the continuity and momentum expressions using `diff`. For instance, in 2D:

  ```
  maxima
  CopyEdit
  (%i3) cont_eq: diff(u(x,y,t),x) + diff(v(x,y,t),y);  /* divergence */
  (%i4) mom_x: rho*(diff(u(x,y,t),t)
                   + u(x,y,t)*diff(u(x,y,t),x)
                   + v(x,y,t)*diff(u(x,y,t),y))
           + diff(p(x,y,t),x)
           - mu*(diff(u(x,y,t),x,2) + diff(u(x,y,t),y,2));
  (%i5) mom_y: rho*(diff(v(x,y,t),t)
                   + u(x,y,t)*diff(v(x,y,t),x)
                   + v(x,y,t)*diff(v(x,y,t),y))
           + diff(p(x,y,t),y)
           - mu*(diff(v(x,y,t),x,2) + diff(v(x,y,t),y,2));
  ```

These expressions represent the PDEs symbolically (LHS minus RHS). We will use them in later days to substitute candidate solutions and check incompressibility.

Day 2: Parametric Coordinates and Velocity/Pressure Fields

- **Parametric mapping.** Treat (u,v,w) as Lagrangian or curvilinear parameters. Define physical coordinates as functions $x=x(u,v,w,t)$, $y=y(u,v,w,t)$, $z=z(u,v,w,t)$. The physical velocity is then $u=(\partial_t x,\partial_t y,\partial_t z)$. For example, a rotating flow might be defined by:

  ```
  maxima
  CopyEdit
  (%i6) x(u,v,t):= u*cos(omega*t) - v*sin(omega*t);
  (%i7) y(u,v,t):= u*sin(omega*t) + v*cos(omega*t);
  ```

  Then compute the velocity field by time derivatives:

  ```
  maxima
  CopyEdit
  (%i8) u_phys(u,v,t) := diff(x(u,v,t),t);  /* velocity component in x */
  (%i9) v_phys(u,v,t) := diff(y(u,v,t),t);  /* velocity component in y */
  ```

  Any scalar field (e.g. pressure) can similarly be defined as $p(x(u,v,t),y(u,v,t),t)$. Working in parametric form allows chain rule transformations if needed for derivative calculations.

- **Divergence-free representations.** A divergence-free vector field in 2D can be expressed via a stream function ψ as $u=\partial_y\psi$, $v=-\partial_x\psi$, which guarantees $\partial_x u+\partial_y v=0$math.ucr.edu. In wxMaxima one can set e.g.

  ```
  maxima
  CopyEdit
  (%i10) psi(x,y):= something;       /* define stream function */
  (%i11) u(x,y) := diff(psi(x,y),y);
  (%i12) v(x,y) := -diff(psi(x,y),x);
  ```

  In 3D one would use a vector potential A with $u=\nabla\times A$math.ucr.edu. These parametric representations automatically enforce $\nabla\cdot u=0$.

Day 3: Divergence-Free Test Fields and Substitution

- **Choose divergence-free test flow.** A common analytical solution is the Taylor–Green vortex (a decaying 2D vortex flow)[en.wikipedia.org](en.wikipedia.org). For example, set

  $u(x,y,t)=\cos(x)\sin(y)\exp(-2\mu t), v(x,y,t)=-\sin(x)\cos(y)\exp(-2\mu t),$

  which is divergence-free (since $\partial_x u+\partial_y v=0$) and satisfies NS with an appropriate pressure. In Maxima:

  ```
  maxima
  CopyEdit
  (%i13) u(x,y,t):= cos(x)*sin(y)*exp(-2*mu*t);
  (%i14) v(x,y,t):= -sin(x)*cos(y)*exp(-2*mu*t);
  ```

- **Verify incompressibility.** Substitute into the continuity expression:

  ```
  maxima
  CopyEdit
  (%i15) cont_test: diff(u(x,y,t),x) + diff(v(x,y,t),y);
  ```

```
(%o15) 0
```

The output confirms zero divergence (symbolically)en.wikipedia.org. Similarly, one can plug u,v,p into the momentum expressions `mom_x`, `mom_y` to compute residuals; if the field is an exact NS solution then these residuals simplify to zero (or to body-forces). This symbolic substitution step checks candidate solutions and generates the PDE residuals algebraically.

Day 4: Symbolic Reynolds Number Re(t)

- **Introduce Re.** In nondimensional form, the Reynolds number Re=UL/ν (ratio of inertial to viscous forces) appears as 1/Re multiplying the viscous termen.wikipedia.org. We can let Re be a time-dependent symbol Re(t) if needed. For example:

```
maxima
CopyEdit
(%i16) Re: Re(t);      /* Reynolds number as a function of time */
(%i17) nu: 1/Re;       /* kinematic viscosity = 1/Re in nondimensional
units */
```

- **Update equations.** Replace the viscosity μ=νρ or simply multiply the viscous term by ν=1/Re. In the momentum expressions, one might factor out ν. For example:

```
maxima
CopyEdit
(%i18) mom_x_nd: rho*(diff(u(x,y,t),t) + u(x,y,t)*diff(u(x,y,t),x)
                        + v(x,y,t)*diff(u(x,y,t),y))
              + diff(p(x,y,t),x)
              - rho*(1/Re)*(diff(u(x,y,t),x,2) + diff(u(x,y,t),y,2));
```

This treats Re symbolically. You can then differentiate or expand these expressions in Maxima, and Maxima will treat `Re(t)` and its time derivative symbolically. For instance, `diff(1/Re(t),t)` yields -Re'(t)/Re(t)^2`. This allows analysis of unsteady or variable-\(Re flows.

Day 5: Numeric Evaluation at Sample Points

- **Substitute sample values.** To check or visualize symbolic results, substitute specific points (x,y,t) into your expressions. For example, take x=1.0,y=2.0,t=0.5:

```
maxima
CopyEdit
(%i19) sub_values: u(1.0,2.0,0.5);
```

or evaluate divergence:

```
maxima
CopyEdit
(%i20) ev(diff(u(x,y,t),x)+diff(v(x,y,t),y), [x=1.0,y=2.0,t=0.5]);
```

This yields a numeric answer (e.g. 0 if divergence-free) and helps verify specific cases. Maxima's `ev(..., [x=..])` or `subst` commands can do this. You can also assign numeric values to parameters like `mu:0.01, rho:1.0` before evaluation for concrete flow checks.

- **Monitor symbolic simplification.** Use `ratsimp()` or `factor()` on complicated expressions after substitution to simplify. For example:

```
maxima
CopyEdit
(%i21) residual_x: ratsimp(ev(mom_x, [x=1.0,y=2.0,t=0.5, mu=0.01,
rho=1.0]));
```

  Ideally, an exact solution yields zero residual (modulo numerical rounding).

Day 6: Diagnostics and Verification

- **Divergence check.** Ensure $\nabla\cdot u=0$ for your symbolic velocity. Compute

```
maxima
CopyEdit
(%i22) div_expr: ratsimp(diff(u(x,y,t),x) + diff(v(x,y,t),y));
```

  and simplify. For our test flows, we found `div_expr = 0` identicallyen.wikipedia.org. If not zero, adjust your fields. This is the key incompressibility diagnostic.

- **Conservation of mass/momentum.** Similarly, verify (symbolically) that `mom_x=0` and `mom_y=0` (with substitutions for p or known body forces) for steady solutions. If using the streamfunction approach, continuity holds by constructionmath.ucr.edu, but momentum must be checked. This can be done by substituting the chosen u,v,p into `mom_x`/`mom_y` and simplifying: e.g.

```
maxima
CopyEdit
(%i23) mom_x_res: ratsimp(mom_x),  mom_y_res: ratsimp(mom_y);
```

  You may need to supply the corresponding pressure gradient (which often balances nonlinear terms). Successful residuals of 0 (or constant) indicate a correct solution.

Day 7: Visualizing Fields – Vector and Scalar Plots

- **Plotting vector fields (draw package).** Load the `draw` package (`load(draw)$`) to use 2D/3D plotting. For example, to plot the 2D velocity field u(x,y), use `draw2d` with `vector` objectsgkerns.people.ysu.edu. A snippet from G. Jay Kerns's tutorial illustrates this approach:

```
maxima
CopyEdit
(%i24) load(draw)$
(%i25) vects: makelist(vector([x_i,y_j],[u(x_i,y_j),v(x_i,y_j)]),
                     x_i, 0, 2*%pi, 0.5,
                     y_j, 0, 2*%pi, 0.5);
(%i26) draw2d(color=blue, head_length=0.1, apply(vects,draw2d,[],
                    gr2d(vects)));
```

  This creates a vector-field plot (arrows) over a grid. As an example, the Taylor–Green vortex flow has the following 2D vector pattern (arrows indicate velocity)en.wikipedia.org:

Figure: 2D velocity field of the Taylor–Green vortex (arrows)[en.wikipedia.org](en.wikipedia.org).

- **Plotting scalar fields (magnitude/pressure).** To plot a scalar field (e.g. velocity magnitude u2+v2 or pressure p(x,y)), use `draw3d(explicit(...))`. For instance, using the Taylor–Green example:

```
maxima
CopyEdit
(%i27) u_mag(x,y) := sqrt(u(x,y)^2 + v(x,y)^2);
(%i28) draw3d(
          colorbox = "Velocity Magnitude",
          explicit(u_mag(x,y), x, 0, 2*%pi, y, 0, 2*%pi)
        );
```

  The `colorbox` and `enhanced3d` options can add a legend[maxima.sourceforge.io](maxima.sourceforge.io). As the manual demonstrates, one can do:

```
maxima
CopyEdit
(%i29) draw3d(
           colorbox="Magnitude", enhanced3d=true,
           explicit(x^2+y^2, x, -1, 1, y, -1, 1)
        );
```

  (This plots x2+y2 as a reference example from the Maxima manual[maxima.sourceforge.io](maxima.sourceforge.io).)

- **Example:** The vector field above suggests pressure contours. One can similarly plot pressure (if known) or derived quantities like vorticity. For example:

```
maxima
CopyEdit
(%i30) p_contour(x,y) := cos(x)*cos(y)*exp(-2*mu*t);  /* sample form */
(%i31) draw2d(
           colorbox="Pressure",
           contour = 10,
           explicit(p_contour(x,y), x, 0, 2*%pi, y, 0, 2*%pi)
        );
```

  which would display pressure contours. The Minima manual contains similar `contour` examples for scalar fields.

- **Exporting plots.** The `draw` package uses gnuplot; figures can be output to screen or files (PNG, PDF, etc.) by setting the `terminal` option (e.g. `terminal="png"`, `terminal="pdf"` in `draw(...)`). For example:

```
maxima
CopyEdit
(%i32) draw2d(terminal="png", filename="velocity_field.png",
             vects);
```

  saves the vector field image.

References

- Incompressible NS equations and divergence-free condition[en.wikipedia.org](en.wikipedia.org)[hplgit.github.io](hplgit.github.io).
- Divergence free velocity via stream functions/vector potentials[math.ucr.edu](math.ucr.edu).

- Taylor–Green vortex exact solutionen.wikipedia.org.
- Maxima `draw` package usage for vector and scalar plotsgkerns.people.ysu.edumaxima.sourceforge.io.
- Reynolds number definition and roleen.wikipedia.orgen.wikipedia.org.