

```

> restart;
with(PDEtools);
[CanonicalCoordinates, ChangeSymmetry, CharacteristicQ, ConservedCurrentTest,
ConservedCurrents, D_Dx, DeterminingPDE, Eta_k, Euler, FromJet,
InfinitesimalGenerator, Infinitesimals, IntegratingFactorTest, IntegratingFactors,
InvariantSolutions, InvariantTransformation, Invariants, Laplace, PDEplot, ReducedForm,
SimilaritySolutions, SimilarityTransformation, SymmetryTest, SymmetryTransformation,
TWSolutions, ToJet, build, casesplit, charstrip, dchange, dcoeffs, declare, diff_table,
difforder, dpolyform, dsubs, mapde, separability, splitstrip, splitsys, undeclare]

```

(1)

## von Neumann stability analysis

The purpose of this worksheet is to introduce a few different stencils for the solution of the diffusion equation and to study their stability properties using the von Neumann stability analysis. We will use the centered\_stencil and onesided\_stencil procedures developed elsewhere:

```

> centered_stencil := proc(r,N,{direction := spatial})
  local n, stencil, vars, beta_sol;
  n := floor(N/2);
  if (direction = spatial) then:
    stencil := D[2$r](u)(t,x) - add(beta[i]*u(t,x+i*h),i=-n..
n);
    vars := [u(t,x),seq(D[2$i](u)(t,x),i=1..N-1)];
  else:
    stencil := D[1$r](u)(t,x) - add(beta[i]*u(t+i*h,x),i=-n..
n);
    vars := [u(t,x),seq(D[1$i](u)(t,x),i=1..N-1)];
  fi:
  beta_sol := solve([coeffs(collect(convert(series(stencil,h,
N),polynom),vars,'distributed'),vars)]:
  stencil := subs(beta_sol,stencil);
  if (direction = spatial) then:
    convert(stencil = convert(series(stencil,h,N+2),polynom),
diff);
  else:
    subs(h=s,convert(stencil = convert(series(stencil,h,N+2),
polynom),diff));
  fi:
end proc:

onesided_stencil := proc(r,N,{direction := spatial})
  local stencil, vars, beta_sol;
  if (direction = spatial) then:
    stencil := D[2$r](u)(t,x) - add(beta[i]*u(t,x+i*h),i=0..
N-1);
    vars := [u(t,x),seq(D[2$i](u)(t,x),i=1..N-1)];
  else:
    stencil := D[1$r](u)(t,x) - add(beta[i]*u(t+i*h,x),i=0..
N-1);
    vars := [u(t,x),seq(D[1$i](u)(t,x),i=1..N-1)];
  fi:
  beta_sol := solve([coeffs(collect(convert(series(stencil,h,
N),polynom),vars,'distributed'),vars)]:
  stencil := subs(beta_sol,stencil);
  if (direction = spatial) then:

```

```

    convert(stencil = convert(series(`leadterm`(stencil), h,
N+1), polynomial), diff);
    else:
    subs(h=s, convert(stencil = convert(series(`leadterm`
(stencil), h, N+1), polynomial), diff));
    fi:
end proc:

```

Using these procedures, we generate three specific stencils. The first two discretize a first order time derivative, while the last discretized a second order spatial derivative.

```

> forward_time := onesided_stencil(1,2,direction=temporal);
backward_time := subs(s=-s, forward_time);
centered_space := centered_stencil(2,3,direction=spatial);

```

$$forward\_time := \frac{\partial}{\partial t} u(t, x) + \frac{u(t, x)}{s} - \frac{u(t + s, x)}{s} = -\frac{1}{2} \left( \frac{\partial^2}{\partial t^2} u(t, x) \right) s \quad (2)$$

$$backward\_time := \frac{\partial}{\partial t} u(t, x) - \frac{u(t, x)}{s} + \frac{u(t - s, x)}{s} = \frac{1}{2} \left( \frac{\partial^2}{\partial t^2} u(t, x) \right) s$$

$$centered\_space := \frac{\partial^2}{\partial x^2} u(t, x) - \frac{u(t, x - h)}{h^2} + \frac{2u(t, x)}{h^2} - \frac{u(t, x + h)}{h^2} = -\frac{1}{12} \left( \frac{\partial^4}{\partial x^4} u(t, x) \right) h^2$$

We will use these to develop a finite difference scheme for the diffusion problem:

```

> pde := diff(u(t,x), t) - d*diff(u(t,x), x, x);

```

$$pde := \frac{\partial}{\partial t} u(t, x) - d \left( \frac{\partial^2}{\partial x^2} u(t, x) \right) \quad (3)$$

Our first scheme involves using the forward\_time and centered space stencils. This gives the FTCS stencil we have previously considered.

```

> FTCS := subs(isolate(lhs(forward_time), diff(u(t,x), t)), isolate
(lhs(centered_space), diff(u(t,x), x, x)), pde);

```

$$FTCS := -\frac{u(t, x)}{s} + \frac{u(t + s, x)}{s} - d \left( \frac{u(t, x - h)}{h^2} - \frac{2u(t, x)}{h^2} + \frac{u(t, x + h)}{h^2} \right) \quad (4)$$

If we instead use the backward\_time stencil for the time derivative, we obtain the BTCS stencil.

```

> BTCS := subs(isolate(lhs(backward_time), diff(u(t,x), t)), isolate
(lhs(centered_space), diff(u(t,x), x, x)), pde);
BTCS := subs(t=t+s, BTCS);

```

$$BTCS := \frac{u(t, x)}{s} - \frac{u(t - s, x)}{s} - d \left( \frac{u(t, x - h)}{h^2} - \frac{2u(t, x)}{h^2} + \frac{u(t, x + h)}{h^2} \right) \quad (5)$$

$$BTCS := \frac{u(t + s, x)}{s} - \frac{u(t, x)}{s} - d \left( \frac{u(t + s, x - h)}{h^2} - \frac{2u(t + s, x)}{h^2} + \frac{u(t + s, x + h)}{h^2} \right)$$

Finally, taking the average of the FTCS and BTCS methods gives the Crank-Nicholson (CN) method.

```

> CN := (FTCS + BTCS)/2;

```

$$(6)$$

$$\begin{aligned}
 CN := & -\frac{u(t, x)}{s} + \frac{u(t+s, x)}{s} - \frac{1}{2} d \left( \frac{u(t, x-h)}{h^2} - \frac{2u(t, x)}{h^2} + \frac{u(t, x+h)}{h^2} \right) \\
 & - \frac{1}{2} d \left( \frac{u(t+s, x-h)}{h^2} - \frac{2u(t+s, x)}{h^2} + \frac{u(t+s, x+h)}{h^2} \right)
 \end{aligned} \tag{6}$$

It is interesting to compare these methods to the forward Euler, backward Euler, and trapezoidal method for solve the following ODE:

```

> ode := diff(y(t), t) = f(t, y(t));
forward_Euler := y(t+s) - y(t) - s*f(t, y(t));
backward_Euler := y(t+s) - y(t) - s*f(t+s, y(t+s));
trapezoidal := y(t+s) - y(t) - s*(f(t, y(t))+f(t+s, y(t+s)))/2;

```

$$\begin{aligned}
 ode := & \frac{d}{dt} y(t) = f(t, y(t)) \\
 forward\_Euler := & y(t+s) - y(t) - s f(t, y(t)) \\
 backward\_Euler := & y(t+s) - y(t) - s f(t+s, y(t+s)) \\
 trapezoidal := & y(t+s) - y(t) - \frac{1}{2} s (f(t, y(t)) + f(t+s, y(t+s)))
 \end{aligned} \tag{7}$$

We see that the FTCS, BTCS, and CN methods are the PDE generalizations of the forward Euler, backward Euler, and trapezoidal methods. We can obtain the leading order error terms in each of the PDE methods by expanding each stencil in a power series in s and h and substituting in the original PDE.

```

> pde_sub := isolate(pde, diff(u(t, x), t));
error_FTCS := convert(convert(dsubs(pde_sub, convert(series(series(
FTCS, s, 3), h, 6), diff)), polynomial), polynomial);
error_BTCS := convert(convert(dsubs(pde_sub, convert(series(series(
BTCS, s, 3), h, 6), diff)), polynomial), polynomial);
error_CN := convert(convert(dsubs(pde_sub, convert(series(series(
CN, s, 4), h, 6), diff)), polynomial), polynomial);

```

$$\begin{aligned}
 pde\_sub := & \frac{\partial}{\partial t} u(t, x) = d \left( \frac{\partial^2}{\partial x^2} u(t, x) \right) \\
 error\_FTCS := & \frac{1}{2} d^2 \left( \frac{\partial^4}{\partial x^4} u(t, x) \right) s - \frac{1}{12} d \left( \frac{\partial^4}{\partial x^4} u(t, x) \right) h^2 \\
 error\_BTCS := & -\frac{1}{2} d^2 \left( \frac{\partial^4}{\partial x^4} u(t, x) \right) s + \left( -\frac{1}{12} d \left( \frac{\partial^4}{\partial x^4} u(t, x) \right) - \frac{1}{12} d^2 \left( \frac{\partial^6}{\partial x^6} u(t, \right. \right. \\
 & \left. \left. x) \right) s \right) h^2 \\
 error\_CN := & -\frac{1}{12} d^3 \left( \frac{\partial^6}{\partial x^6} u(t, x) \right) s^2 + \left( -\frac{1}{24} d^2 \left( \frac{\partial^6}{\partial x^6} u(t, x) \right) s - \frac{1}{12} d \left( \frac{\partial^4}{\partial x^4} u(t, \right. \right. \\
 & \left. \left. x) \right) - \frac{1}{48} d^3 \left( \frac{\partial^8}{\partial x^8} u(t, x) \right) s^2 \right) h^2
 \end{aligned} \tag{8}$$

Since the error in the FTCS stencil is  $O(s) + O(h^2)$ , we call that method first order in time and second order in space, as is the BTCS method. But the CN stencil is second order in time and second order in

space. Now, if we define the error E to be the difference between a numeric and true solutions for u(t,x), then it follows that E(t,x) obeys the following for each stencil:

```
> FTCS_E := subs(u=E, FTCS);
   BTCS_E := subs(u=E, BTCS);
   CN_E := subs(u=E, CN);
```

$$FTCS\_E := -\frac{E(t,x)}{s} + \frac{E(t+s,x)}{s} - d \left( \frac{E(t,x-h)}{h^2} - \frac{2E(t,x)}{h^2} + \frac{E(t,x+h)}{h^2} \right) \quad (9)$$

$$BTCS\_E := \frac{E(t+s,x)}{s} - \frac{E(t,x)}{s} - d \left( \frac{E(t+s,x-h)}{h^2} - \frac{2E(t+s,x)}{h^2} + \frac{E(t+s,x+h)}{h^2} \right)$$

$$CN\_E := -\frac{E(t,x)}{s} + \frac{E(t+s,x)}{s} - \frac{1}{2} d \left( \frac{E(t,x-h)}{h^2} - \frac{2E(t,x)}{h^2} + \frac{E(t,x+h)}{h^2} \right) - \frac{1}{2} d \left( \frac{E(t+s,x-h)}{h^2} - \frac{2E(t+s,x)}{h^2} + \frac{E(t+s,x+h)}{h^2} \right)$$

The von Neumann stability analysis assumes the following ansatz for the error:

```
> E := (t,x) -> xi^(t/s)*exp(I*k*x);
```

$$E := (t,x) \rightarrow \xi^{\frac{t}{s}} e^{kx} \quad (10)$$

Putting this into each of E equations of motion and solving for the **amplification factor** xi, we obtain

```
> xi_FTCS := simplify(isolate(convert(simplify(expand(FTCS_E/E(t,x)), power), trig), xi));
   xi_BTCS := simplify(isolate(convert(simplify(expand(BTCS_E/E(t,x)), power), trig), xi));
   xi_CN := simplify(isolate(convert(simplify(expand(CN_E/E(t,x)), power), trig), xi));
```

$$xi\_FTCS := \xi = \frac{h^2 + 2ds \cos(kh) - 2ds}{h^2} \quad (11)$$

$$xi\_BTCS := \xi = -\frac{h^2}{-h^2 + 2ds \cos(kh) - 2ds}$$

$$xi\_CN := \xi = -\frac{h^2 + ds \cos(kh) - ds}{-h^2 + ds \cos(kh) - ds}$$

A particular method will be unstable if |xi|>0, as this implies that the error are growing exponentially in time. To determine under which circumstances this holds, we first simplify things by making the following definitions:

```
> Subs := [s = -z*h^2/d/4, k=q/h];
```

$$Subs := \left[ s = -\frac{z h^2}{4 d}, k = \frac{q}{h} \right] \quad (12)$$

We arrange our solutions for xi into a list and make the above substitutions

```
> xi_sols := simplify(subs(Subs, [xi_FTCS, xi_BTCS, xi_CN]));
```

$$xi\_sols := \left[ \xi = 1 - \frac{1}{2} z \cos(q) + \frac{z}{2}, \xi = \frac{2}{2 + z \cos(q) - z}, \xi = -\frac{-4 + z \cos(q) - z}{4 + z \cos(q) - z} \right] \quad (13)$$

The extreme values of these are obtained when  $\cos(q) = +1$  or  $-1$ . Putting those values in

```
> extreme_xi_1 := subs(cos(q)=+1,xi_sols);
extreme_xi_2 := simplify(subs(cos(q)=-1,xi_sols));
extreme_xi_1 := [\xi = 1, \xi = 1, \xi = 1]
```

(14)

$$extreme\_xi\_2 := \left[ \xi = 1 + z, \xi = -\frac{1}{-1 + z}, \xi = -\frac{2 + z}{-2 + z} \right]$$

It is clear that `extreme_xi_1` does not give us any useful information, but `extreme_xi_2` does. We can now enforce the stability criterion  $|xi| < 1$ :

```
> stability_criteria_1 := map(x->subs(x,abs(xi)<1),extreme_xi_2);
stability_criteria_1 := \left[ |1 + z| < 1, \left| -\frac{1}{-1 + z} \right| < 1, \left| -\frac{2 + z}{-2 + z} \right| < 1 \right]
```

(15)

Notice these are exactly the same form as the stability criteria for the forward, backward and trapezoidal scheme for ODEs. In terms of the original parameters of the stencils, the stability criteria read:

```
> stability_criteria_2 := map(x->{rhs(x)<1,rhs(x)>-1},simplify(subs(isolate(Subs[1],z),extreme_xi_2)));
stability_criteria_2 := \left[ \left\{ -1 < -\frac{-h^2 + 4 d s}{h^2}, -\frac{-h^2 + 4 d s}{h^2} < 1 \right\}, \left\{ -1 < \frac{h^2}{h^2 + 4 d s}, \frac{h^2}{h^2 + 4 d s} < 1 \right\}, \left\{ -1 < -\frac{-h^2 + 2 d s}{h^2 + 2 d s}, -\frac{-h^2 + 2 d s}{h^2 + 2 d s} < 1 \right\} \right]
```

(16)

These can be solved by MAPLE to give explicit conditions on  $s$  (assuming  $d$  is positive):

```
> stability_criteria_3 := map(x->solve(x,s),stability_criteria_2)
assuming(s>0,d>0,h>0);
stability_criteria_3 := \left[ \left\{ 0 < s, s < \frac{h^2}{2 d} \right\}, \{0 < s\}, \{0 < s\} \right]
```

(17)

The first element in the list says that the FTCS scheme is stable if  $s$  is less than  $h^2/2d$ , while the second and third elements say that the BTCS and CN methods are unconditionally stable.