Fox Lecture

~~Caltech Physics~~ Notes

Ph129 - A Differential

Equations

INCOMPLETE

$$X[i+1] = X[i] + h*\{F(X*, t[i+1]) + F[i]\}/2$$

Example 2: 4th-order Adams-Bashforth/Adams-Moulton

$$X* = X[i] + h*\{55*F[i] - 59*F[i-1] + 37*F[i-2] - 9*F[i-3]\}/24$$

$$X[i+1] = X[i] + h*\{9*F(X*, t[i+1]) + 19*F[i] - 5*F[i-1] + F[i-2]\}/24$$

- Generally, Adams-Bashforth/Adams-Moulton methods of the same order make good predictor/corrector pairs

Why we want to use implicit methods:

- All of these methods are trying to approximate

$$X[i+1] - X[i] = \int_{u=t[i]}^{t[i+1]} X'(u)\, du = \int_{u=t[i]}^{t[i+1]} F(X(u), u)\, du$$

- Explicit method:

```
   |<-------------- known --------------->|<-unknown-->|

 F(X,t)                                          ******
        ********F**                          ******/////|
     ***        |  *****              *****F*///////////|
    *           |       ***     *****     |/////////////|
   F            |          **F**          |/////////////|
   |            |            |            |/////////////|
   +------------+------------+------------+------------+
t[i-3]       t[i-2]       t[i-1]        t[i]        t[i+1]
                                   Shaded area gives increment
```

This is extrapolation!

- Implicit method: { X[i+1] is treated as though known. }

```
   |<-------------- known --------------->|<-"known"-->|

 F(X,t)                                          *****F
        ********F**                          ******/////|
     ***        |  *****              *****F*///////////|
    *           |       ***     *****     |/////////////|
   F            |          **F**          |/////////////|
   |            |            |            |/////////////|
   +------------+------------+------------+------------+
t[i-3]       t[i-2]       t[i-1]        t[i]        t[i+1]
                                   Shaded area gives increment
```

This is interpolation, inherently superior!

Solving the implicit equation more accurately

- Equation has the form

$$Z = G(Z)$$

- We have a good initial approximation call it Z[0]

- Calculate successive approximations

$$Z[k+1] = G(Z[k])$$

Theorem: If G has a Lipschitz constant $M < 1$ such that

$$\| G(Z) - G(Y) \| <= M*\| Z - Y \|$$

for any vectors Z, Y, then the iteration above converges to the unique solution of $Z = G(Z)$.

In fact, each iteration reduces the error by a factor of M.

Correcting more than once:

- The iteration will always work for sufficiently small h:
  - F has a Lipschitz constant L, possibly large
  - The equation (using Z for X[i+1]) is

$$Z = X[i] + h*\{9*F(Z, t[i+1]) + 19*F[i] - 5*F[i-1] + F[i-2]\}/24$$

  - The Lipschitz constant for "G" is $9*h*L/24$.
- It is usually felt that a single correction is sufficient, but such may not always be the case.

Monitoring error with predictor/corrector methods

Consider 3rd order Adams-Bashforth/Adams-Moulton

$$Y = \text{exact solution for } t[i+1] \text{ starting from } X[i] \text{ at } t[i]$$

$$X^* = X[i] + h*\{23*F[i] - 16*F[i-1] + 5*F[i-2]\}/12$$

$$= Y + (3/8)*h^4*f^(3)$$

$$X[i+1] = X[i] + h*\{5*F(X^*,t[i+1]) + 8*F[i] - F[i-1]\}/12$$

$$= Y - (1/24)*h^4*f^(3)$$

f^(3) is evaluated at different places in the error terms, but with error O(h^5) we can write

$$X^* - X[i+1] = (10/24)*h^4*f^(3)$$

$$X[i+1] \sim= Y - (1/10)*(X^* - X[i+1])$$

We could also extrapolate using

$$Y = (1/10)*X^* + (9/10)*X[i+1] + O(h^5)$$

All this assumes that X[i+1] is a good solution of the corrector, if in doubt, perform additional corrections before testing error.

Getting started

- Use a one-step method of the same order as the multistep method,

e.g., a Runge-Kutta method.

- Take the time step smaller than planned for continuing the calculation.

References (in addition to those for the course generally):

P. Henrici. Discrete Variable Methods in Ordinary Differential Equations. John Wiley & Sons, Inc., New York, 1962.

A. Marciniak. Numerical Solutions of the N-body Problem. D. Reidel Publishing Co., Dordrecht, 1985.

D. Greenspan. Arithmetic Applied Mathematics. Pergamon Press (International Series in Nonlinear Mathematics), Oxford, 1980.

R.A. LaBudde and D. Greenspan. Energy and Momentum Conserving Methods of Arbitrary Order for the Numerical Integration of Equations of Motion. Numerische Mathematik, vol. 26, pp. 1-16, 1976.

ode2.notes

Ordinary Differential Equations

Part 2

Stability of multi-step methods

Methods considered so far (explicit or implicit) have the form

```
X[i+1] = X[i] + h*{...}
```

Consider an implicit method due to Milne:

```
X[i+1] = X[i-1] + h*{ F[i+1] + 4*F[i] + F[i-1] }/3
```

(Simpson's rule as a corrector, in effect)

As h --> 0 we obtain a limiting difference equation:

```
X[i+1] - X[i] = 0                A-B, A-M

X[i+1] + 0*X[i] - X[i-1] = 0     Milne
```

and similarly for other multi-step methods

Homogeneous difference equations with constant coefficients:

For the equation

```
a[k]*x[i+k] + a[k-1]*x[i+k-1] + . . . + a[0]*x[i] = 0
```

consider solutions of the form x[i] = z^i (z is some number), so

```
a[k]*z^(i+k) + a[k-1]*z^(i+k-1) + . . . + a[0]*z^(i) = 0

          a[k]*z^(k) + a[k-1]*z^(k-1) + . . . + a[0] = 0
```

So roots of p(z) = a[k]*z^(k) + a[k-1]*z^(k-1) + . . . + a[0] generate solutions.

If u is a root of p(z) of multiplicity q > 1 then all of the following are solutions:

```
x[i] = u^i

x[i] = i*u^i

     :

x[i] = i*(i - 1)*(i - q + 2)*u^i
```

Stability condition:

For a convergent multi-step method, all of the (possibly complex) roots of the associated polynomial p(z) must satisfy

```
|u| <= 1
```

and if |u| = 1 then u must be simple.

If this condition fails, the difference method has erroneous components

which become large as h --> 0.

Weak stability:

All of the methods discussed are stable. Milne's method, however has

$$u = 1, \quad u = -1$$

as roots, hence the oscillatory solution

$$x[i] = (-1)^i$$

$$1, -1, 1, -1, 1, -1, \ldots$$

which can cause trouble if the solution is rapidly decreasing.

Milne's method is "weakly stable".

Strong stability:

$p(1) = 0, \quad |u| < 1$ for all roots $u \neq 1$.

Strongly stable methods (e.g., A-B, A-M) are best for general use.

Boundary Value Problems

Consider, as a slightly special case,

$$x''(t) = f(x, t), \quad a <= t <= b$$

$$x(a) = xa, \quad x(b) = xb$$

x is a scalar function of t.

Recall that

$$x''(t) = \frac{x(t+h) - 2*x(t) + x(t-h)}{h^2} + O(h^2)$$

For $h = (b - a)/n$ and $t[i] = a + i*h$ we obtain approximations $x[i] \sim= x(t[i])$ from the equations

$$\begin{aligned}
2*x[1] - x[2] \qquad\qquad\qquad\qquad &+ h^2*f[1] = xa \\
- x[1] + 2*x[2] - x[3] \qquad\qquad &+ h^2*f[2] = 0 \\
&\vdots \\
-x[i-1] + 2*x[i] - x[i+1] \qquad &+ h^2*f[i] = 0 \\
&\vdots \\
-x[n-2] + 2*x[n-1] &+ h^2*f[n-1] = xb
\end{aligned}$$

where $f[i] = f(x[i], t[i])$.

f depends only on t

We have a tri-diagonal linear system

$$A\ X = B - h^2*F$$

i.e.,

$$
\begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & & \ddots & & \\ & & & \ddots & \\ & & & \ddots & \\ & 0 & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}
\begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ \vdots \\ \vdots \\ x[n-2] \\ x[n-1] \end{bmatrix}
= \begin{bmatrix} xa \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ xb \end{bmatrix}
- h^2* \begin{bmatrix} f[1] \\ f[2] \\ \vdots \\ \vdots \\ \vdots \\ f[n-2] \\ f[n-1] \end{bmatrix}
$$

which can be solved directly.

f depends on x and t

- Solve non-linear system iteratively
- Requires repeated solution of tri-diagonal linear system
  - Direct solution requires effort O(n) per iteration
  - Tricky to perform in parallel
  - Parallel iterative solution of tri-diagonal system might be satisfactory.
- Higher-order appoximations to x'' might be used, in which case the matrix A will have more non-zero off-diagonal elements

Eigenvalue Problems

Another type of boundary-value problem:

$$x''(t) + \text{lambda}*q(t)*x(t) = 0, \quad a <= t <= b$$

$$x(a) = x(b) = 0, \quad q(t) > 0 \text{ for } a <= t <= b$$

for which we wish to determine both lambda (a scalar) and x.

Discretize as for the standard boundary-value problem:

$$A\ X + h^2*\text{lambda}*Q\ X = 0$$

$$\text{where } Q = \begin{bmatrix} q(t[1]) & & & & \\ & q(t[2]) & & 0 & \\ & & \ddots & & \\ & & & \ddots & \\ & 0 & & & q(t[n-1]) \end{bmatrix}$$

Rewrite the matrix equation as

$$Q^{(-1)}\ A\ X + h^2*\text{lambda}*X = 0$$

or

$$Q^{(-1)}\ A\ X = -h^2*\text{lambda}*X$$

to obtain a standard, matrix eigenvalue problem.

auto.notes

Automatic Step-size Adjustment for One-step Methods

Press, et.al., treat this subject in detail. These notes are based largely on their discussion.

In general we want

Large steps when solution is changing slowly

Small steps when solution is changing rapidly

Using a fixed step h throughout is likely to be inefficient.

Basic idea

On each time step:

Integrate from t to t+h using step h and also h/2

Estimate error from the two results

If error too large, determine reduced h, repeat

Otherwise, extrapolate for result of this step

estimate h for next step

- Quite possibly no two steps use the same h.
- Accuracy should be a little better than that corresponding to a step h/2 due to extrapolation.
- Effort is less than 1.5 times effort for step h/2, since F(X, t) is used twice.

Details

Suppose method is M(t,X,h), with local error O(h^m). To advance one step from

t = t0, X(t) = X0

proceed as follows:

Calculate:

```
X1 = X0 + h*M(t0, X0, h)            { one step of h }

X* = X0 + h*M(t0, X0, h/2)
                                    { two steps of h/2 }
X2 = X* + h*M(t0+h/2, X*, h/2)
```

If Z is the exact solution starting from X0 then

```
Z = X1 + 2^m*h^m*e + O(h^(m+1))

Z = X2 +   2*h^m*e + O(h^(m+1))
```

$$Z = X2 + (X2 - X1)/(2^{(m-1)} - 1) + O(h^{(m+1)})$$

The last equation is the extrapolation formula. It is tempting to estimate the error from

$$(X2 - X1)/(2^{(m-1)} - 1)$$

but this is risky because of $O(h^{(m+1)})$. The customary, and safer, method is to use

$$(X2 - X1)$$

as the error indicator. We consider how to use this a little later.

Adjusting h

If E0 is the error we want, and E1 the error obtained with step h1, then the step which yields error E0 is

$$h0 = h1 \left| \frac{E0}{E1} \right|^{(1/m)} \qquad \text{(approximately)}$$

Let h1 = 'h' value just used, E1 = corresponding error

If E1 > E0 the step fails, repeat with h = h0

If E1 <= E0 accept this step, calculate h0 as estimate for next step.

Determining the error

We might use any of several criteria, depending on the behavior of the components of X:

- Relative error (components of very different magnitudes)
- Absolute error (bounded components, perhaps oscillating)
- Different requirements for different components
- May wish to control global error, not just error/step

General scheme:

User gives:

eps - overall tolerance, eps > 0 .

XSC - "scale" vector, same shape as X .

Goal:

$$|X1 - X0| <= eps*|XSC|$$

Error measure:

$$E(X1 - X0) = \text{maxval}(\ |(X1 - X0)/XSC| \ )$$

Choice of eps, XSC

Relative error: eps fixed, XSC = X

Absolute error: eps fixed, XSC fixed

Relative except when X near 0:

eps fixed, $XSC = |X| + h*|F(X, t)| + mu$

mu is a microscopic constant

Global error: $eps = epsg*h$, $YSC = F(X, t)$,

where epsg is the global error criterion.

Adjusting h when criterion may involve h

Whether shrinking or expanding, the previous equation changes h too little if error requirement is proportional to h.

Better: Write $E1 = E(X2 - X1)$ as above.

$$E1 > eps: \quad h0 = 0.9*h1*\left|\frac{eps}{E1}\right|^{(1/(m-1))}$$

$$E1 < eps: \quad h0 = 0.9*h1*\left|\frac{eps}{E1}\right|^{(1/m)}$$

"0.9" is a somewhat arbitrary safety factor. Both formulas are conservative.

Observations

- t must be calculated by accumulating h's, rounding error may grow.
- h must not be made so small that $fl(t+h) = t$.
  { $fl(t+h)$ is the floating point result for t+h }
- h should not be made so small that $X(t+h) = X(t)$, especially when $F(X, t)$ does not depend explicitly on t.
- When integrating from ta to tb, program must adjust h when $t+h > tb$.

Typical program organization

Top level control:

Inputs: tstart, Xstart = X(tstart), Deltat, nint, eps
{ , hstart, hmin, XSC }

Task: Calculate and report

X(tstart), X(tstart + Delta-t), ..., X(tstart + nint*Deltat)

Deltat is typically much larger than the time step used to integrate the equations.

Adaptive control procedure:

Inputs: ta, tb, Xa, ha, hmin, eps { , XSC }

Task: Integrate from initial conditions ta, Xa to tb

ha is step to try at beginning

hmin is smallest allowed h

eps { and XSC } specify error tolerance

Returns calculated X corresponding to tb, also h0 to use if integration is continued beyond tb.

Reports failure (h < hmin, etc.)

Adaptive stepper:

Inputs: t0, X0, hx, eps, XSC

Task: Perform one time step, trying t0 to t0+hx,

Returns: X1, hf, hnext

X1 is extrapolated solution for t0+hf, hf <= hx

hnext is suggestion for next step

Reports failure if fl(t0+hf) = t0

nbody.notes

The N-body Problem for Newtonian Gravitation

Newton's law of gravitation

The force on P[i] (particle i) due to P[j] is

```
                    *  P[j]
                   /
                  /
                |/ F[j,i]

        F[i,j]  /|
               /
              /
        P[i] *
```

$$F[i,j] = G*m[i]*m[j]*\frac{X[j] - X[i]}{r[i,j]^3}$$

where

X[i] = (x[i], y[i], z[i]) = (vector) position of P[i]

m[i] = mass of P[i]

r[i,j] = || X[j] - X[i] || (Euclidean distance)

G = 0.0001 1856 4121 au-earth-yr (solar system)

  = 4.497 086 Kpsc-Tsun-Myr (galaxies)

Note that this is an inverse-square force. The total force on P[i] is the (vector) sum for i != j of F[i,j].

The equations of motion

For 1 <= i <= N we write

V[i] = dX[i]/dt (velocity of P[i])

A[i] = dV[i]/dt (acceleration of P[i])

whence

$$m[i]*A[i] = G*m[i]*\sum_{\substack{j=1 \\ j!=i}}^{N} m[j]*\frac{X[j] - X[i]}{r[i,j]^3}$$

or

$$A[i] = G*\sum_{j!=i} m[j]*\frac{X[j] - X[i]}{r[i,j]^3}$$

Rather than write everything as a single vector we shall keep positions, velocities, and accelerations separate. We define

```
X = [   X[0],  X[1],   .  .  . , X[N]  ]

    |_ x[0],  x[1],  .  .  . ,  x[N] _|
  = |  y[0],  y[1],  .  .  . ,  y[N]  |
    |_ z[0],  z[1],  .  .  . ,  z[N] _|

Y = [   Y[0],  Y[1],  .  .  . ,  Y[N]  ]

A = [   A[0],  A[1],  .  .  . ,  A[N]  ]  =  Grav(X)
```

where Grav(X) is determined as above. We obtain the equations of motion

```
dX/dt = V,   dV/dt = A = Grav(X) .
```

Molecular forces

The situation is analogous, except for the nature of the force law. Instead of F[i,j] being proportional to 1/r[i,j]^2 we might have

```
                        |_   A            B     _|
F[i,j] = m[i]*m[j]*| -------- - --------- |*(X[j]-X[i])
                        |_r[i,j]^6   r[i,j]^10_|
```

which is highly repulsive for small r, diminishingly attractive for large r.

Standard numerical methods for gravitation

To avoid confusion with particle positions, etc., we write X(k), V(k), A(k) for the (total) position, velocity, and acceleration vectors at time step k, corresponding to t = t0 + k*h.

Standard methods have the form:

```
Y(i+1) = Y(i) + h*{ form in F, Y, t }
```

Here Y = [X,V], F(Y, t) = F([X,V], t) = [V, Grav(X)]

Euler's method: (useless in practice)

```
X(k+1) = X(k) + h*V(k)

V(k+1) = V(k) + h*Grav(X(k))
```

Modified Euler's method: (global order 2)

```
X* = X(k) + h*V(k)/2

V* = V(k) + h*Grav(X(k))/2

X(k+1) = X(k) + h*V*
```

```
            = X(k) + h*V(K) + h^2*Grav(X(k))/2

      V(k+1) = V(k) + h*Grav(X*)
```

Adams-Bashforth/Adams Moulton: (global order 4)

```
      A(k) = Grav(X(k))

      X* = X(k) + h*{  55*V(k)    - 59*V(k-1)

                     + 37*V(k-2) -  9*V(k-3) }/24

      V* = V(k) + h*{  55*A(k)    - 59*A(k-1)

                     + 37*A(k-2) -  9*A(k-3) }/24

      A* = Grav(X*)

      X(k+1) = X(k) + h*{  9*V*      + 19*V(k)

                         - 5*V(k-1) +    V(k-2) }/24

      V(k+1) = V(k) + h*{ 9*A*      + 19*A(k)

                         - 5*A(k-1) +    A(k-2) }/24
```

The constants of motion in Newtonian mechanics

We have no dissipative forces which would convert mechanical energy to heat, so the motion of the particles, if calculated exactly, would conserve energy (kinetic + potential), linear momentum, angular momentum, and would also have the property that the center of mass moves in a straight line at constant speed (the center of mass has constant velocity). In fact, there are constant vectors Cm, Cc, Ca and a scalar constant E such that, at any time t,

```
Linear Momentum:    SUM m[i]*V[i] = Cm
                     i

Center of mass:     SUM m[i]*X[i] = Cm*t + Cc
                     i

Angular momentum:   SUM m[i]*(X[i] x V[i]) = Ca
                     i                       { x is cross product }

Kinetic energy:     (1/2)*SUM m[i]*|| V[i] ||^2 = K(t)
                           i

                          m[i]*m[j]
Potential energy:   G*SUM ---------- = P(t)
                     i<j   r[i,j]

Total Energy:       K(t) + P(t) = E
```

Theorem: Euler's method conserves linear momentum and the velocity of the center of mass (aside from rounding error).

Proof: Obviously,

```
    SUM m[i]*{V[i](k+1) - V[i](k)} = SUM m[i]*h*Grav(X(k))[i]
```

i i

but

$$\mathrm{Grav}(X(k))[i] = G*\sum_{j!=i} m[j]*\frac{X[j] - X[i]}{r[i,j]^3}$$

so when we write everything out, corresponding to

$$m[i]*h*G*m[j]*\frac{X[j] - X[i]}{r[i,j]^3}$$

we also obtain

$$m[j]*h*G*m[i]*\frac{X[i] - X[j]}{r[j,i]^3}$$

and these two cancel, so the complete sum is zero. Thus linear momentum never changes.

Now we also have

$$\sum_i m[i]*(X[i](k+1) - X[i](k)) = h*\sum_i m[i]*V[i](k)$$

$$= h*Cm$$

so the position of the center of mass is Cm*t + Cc .

q. e. d.

Claim: Any reasonable numerical method conserves linear momentum and the velocity of the center of mass.

Argument: The details will be messy, but a similar argument will apply. The velocity increment will be exactly such an expression, or a weighted average of such, so the same cancellation will apply. The position increment is some velocity*h, or a weighted average of such, so the same conclusion will follow.

Energy-conserving methods

We can adapt any one-step method to obtain an energy conserving method, though at considerable computational expense.

Example using the modified Euler method (global order 2):

Notation:
- Z = X, V (the computed system state)
- Y = exact solution integrating from Z[i] for time h
- F(Z) = V, Grav(X)
- E(Z) = total energy from Z
- E0 = E(Z0) (initial energy)

For each time step, calculate:

```
Z1 := Z(k) + h*F( Z(k) + (h/2)*F(Z(k)) )     { mod. Euler }

Z2 := Z(k) + (h/2)*F(Z(k) + (h/4)*F(Z(k)))  { mod. Euler, step h/2 }
```

```
Z2 := Z2 + (h/2)*F(Z2 + (h/4)*F(Z2))          { second step of h/2 }

Z(k+1) = (1 - a)*Z1 + a*Z2
```

where a is determined by solving

```
E( (1 - a)*Z1 + a*Z2 ) = E0
```

for the parameter 'a' using any convenient numerical method (secant, bisection, Newton).

The weight a should be about 4/3; the value should change only slightly from one time step to the next.

Why it works:

From the previous discussion of error we can say that the exact solution Y satisfies

```
Y = Z1 +     h^2*e + c*h^3*U,            ||U|| <= 1

Y = Z2 + (h^2/4)*e + c*(h^3/8)*V         ||V|| <= 1

Y = (1 - a)*Z1 + a*Z2 + (1 - a + a/4)*h^2*e
                      + {(1 - a)*U + a*V/8)*c*h^3

  = Z(k+1) + (1 - 3*a/4)*h^2*e + (|1 - a| + |a|/8)*c*h^3*U'

  where   ||U'|| <= 1
```

but

```
E(Y) = E0
```

In practice we can be sure there will be a reasonable value for 'a' which yields E((1 - a)*Z1 + a*Z2) = E0 .

This technique is not widely discussed in connection with multistep methods, though one could consider using predicted and corrected values as the basis for the extrapolation.

nbring.notes

Calculating accelerations in an N-processor ring

Data:

- N = number of processors (even)
- NB = actual number of particles, <= N
- Fi is "fixed" data for particle i

  Xf - position
  Mf - mass
  Af - accumulating acceleration

- Ci is "circulating" data for particle i

  Xc - position
  Mc - mass
  Ac - accumulating acceleration

Basic step:

Circulate C one step to right

Calculate accelerations both ways, accumulating in Af, Ac

In some circumstance the last step should calculate the acceleration only one way.

Illustrations with N = 8:

NB = 6 > 1/2 N

```
        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|F4|F5|F6|Fx|Fx|
0       +--+--+--+--+--+--+--+--+        Start - no calculation
        |C1|C2|C3|C4|C5|C6|Cx|Cx|
        +--+--+--+--+--+--+--+--+

        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|F4|F5|F6|Fx|Fx|
1       +--+--+--+--+--+--+--+--+        Acceleration both ways
        |Cx|C1|C2|C3|C4|C5|C6|Cx|
        +--+--+--+--+--+--+--+--+
             x  x  x  x  x

        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|F4|F5|F6|Fx|Fx|
2       +--+--+--+--+--+--+--+--+        Acceleration both ways
        |Cx|Cx|C1|C2|C3|C4|C5|C6|
        +--+--+--+--+--+--+--+--+
                x  x  x  x

        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|F4|F5|F6|Fx|Fx|
3       +--+--+--+--+--+--+--+--+        Acceleration both ways
        |C6|Cx|Cx|C1|C2|C3|C4|C5|
        +--+--+--+--+--+--+--+--+
          x         x  x  x

        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|F4|F5|F6|Fx|Fx|
4       +--+--+--+--+--+--+--+--+        Fixed accelerations only
```

```
|C5|C6|Cx|Cx|C1|C2|C3|C4|
+--+--+--+--+--+--+--+--+
  f  f        f  f
```

N/2 - 1 two-way steps, 1 one-way step

Ac is shifted N/2 from starting position

NB = 3 <= 1/2 pipe-length

```
        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|Fx|Fx|Fx|Fx|Fx|
0       +--+--+--+--+--+--+--+--+       Start - no calculation
        |C1|C2|C3|Cx|Cx|Cx|Cx|Cx|
        +--+--+--+--+--+--+--+--+

        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|Fx|Fx|Fx|Fx|Fx|
1       +--+--+--+--+--+--+--+--+       Acceleration both ways
        |Cx|C1|C2|C3|Cx|Cx|Cx|Cx|
        +--+--+--+--+--+--+--+--+
             x  x

        +--+--+--+--+--+--+--+--+
        |F1|F2|F3|Fx|Fx|Fx|Fx|Fx|
2       +--+--+--+--+--+--+--+--+       Acceleration both ways
        |Cx|Cx|C1|C2|C3|Cx|Cx|Cx|
        +--+--+--+--+--+--+--+--+
                x
```

NB - 1 two-way steps

Ac is shifted NB - 1 from starting position

We are using anti-symmetry, but the calculation is very inefficient.

nbmode.fcm

```
C  N-body calculation using the modified Euler method,
C  which is globally 2nd-order.
C  Uses a ring organization to calculate accelerations.
C
C  Data arrays (position, velocity, acceleration, mass)
C  are organized:
C
C          1          Particle number (i) -->          N
C         +-------------------------------------------+
C   |  x  |                                           |
C  (c) y  |                                           |
C   v  z  |                                           |
C         +-------------------------------------------+
C
C  The parameter N and common blocks /NBODYFE/, /NBODYPE/ with arrays
C
C       X0, X1  - Current, next positions
C       V0, V1  - velocities
C        M, GM  - mass, G*mass (redundant)
C        c,  i  - array coordinates
C  and
C           NB  - integer <= N, actual no. of bodies
C            G  - real, the gravitational constant
C
C  are defined in the file nbmode.h
C
C  File:  nbmode.fcm


C------- setup() ---------------------------------------

      subroutine setup()

      INCLUDE 'nbmode.h'

C  Initialize coordinates
      c = SPREAD( [1:3],  2, N)
      i = SPREAD([1:N], 1, 3)
      end subroutine setup


C------- init() ----------------------------------------

C  Load initial positions, velocities, also masses,
C  gravitational const.  All are read from logical
C  unit 10.  Replicates masses over x,y,z.

      subroutine init()

      INCLUDE 'nbmode.h'

      integer k
      real mx
      X0 = 0.0            ! For unused particles
      V0 = 0.0
      do, k = 1, NB
        read (10, *) X0(1, k), X0(2, k), X0(3, k)
      end do
      do, k = 1, NB
        read (10, *) V0(1, k), V0(2, k), V0(3, k)
      end do
      M = 0.0             ! For unused particles
      do, k = 1, NB
        read (10, *) mx
        M(:, k) = mx
      end do
```

```
      read (10, *) G
      end subroutine init


C------- mkast(nx, ar, am) ----------------------------

C  Insert nx asteroids at radius ar with total mass am.
C  This assumes that an orbit of radius 1 about origin
C  corresponds to speed 2*pi (e.g., au-earth-yr units).

      subroutine mkast(nx, ar, am)
      integer nx          ! Number of new asteroids
      real ar, am         ! Orbit radius, total mass

      INCLUDE 'nbmode.h'

      real, parameter:: TWOPI = 6.2831853
      real, array(1:3, 1:N) :: sx, cx, arg
      real spd
      integer nn
      print 21, nx, ar, am
 21   FORMAT (" Making ", I4, " asteroids, radius ",
     *        F10.5, ", mass ", F8.5)
      nn = NB + nx
      spd = TWOPI/sqrt(ar)
      where ( (NB < i) .and. (i <= nn) )
        arg = (TWOPI*(i - NB))/nx
        sx = sin(arg)
        cx = cos(arg)
      end where
      where ( (NB < i) .and. (i <= nn) .and. (c == 1) )
        X0 =  cx
        V0 = -sx
      end where
      where ( (NB < i) .and. (i <= nn) .and. (c == 2) )
        X0 =  sx
        V0 =  cx
      end where
      where ( (NB < i) .and. (i <= nn) .and. (c == 3) )
        X0 = 0.0
        V0 = 0.0
      end where
      where ( (NB < i) .and. (i <= nn) )
        X0 = X0*ar
        V0 = V0*spd
         M = am/nx
      end where
      NB = nn
      end subroutine mkast


C------- print_state(PX) ------------------------------

C  Print current state of system according to PX.
C  See comments for main program for PX settings.

      subroutine print_state(PX)
      integer PX          ! Print control

      INCLUDE 'nbmode.h'

      integer k, np
      np = NB
      if ( (PX >= 0) .and. (PX < NB) ) np = PX
      print 21
 21   FORMAT (/ "          x          y          z      ",
```

```
     *          "    vx         vy         vz         m")
      do, k = 1, np
        print 22, k, X0(1,k), X0(2,k), X0(3,k),
     *               V0(1,k), V0(2,k), V0(3,k), M(1,k)
      end do
 22   FORMAT (1X, I3, 6F10.5, E10.2)
      end subroutine print_state

C------- print_x(ax, na, nf) ------------------------------

C  Print ax(:, na:nf), nf-na <= 9 (for testing)

      subroutine print_x(ax, na, nf)
      INCLUDE 'nbmode.h'
      real, array(1:3, 1:N) :: ax
CMPF  ONDPU ax
CMF$  LAYOUT ax(,)
      integer na, nf

      integer u, lm
      lm = min(nf, na+9)
      do, u = 1, 3
        print 21, ax(u, na:lm)
 21     FORMAT ( 10(1X : E9.2 :) )
      end do
      end subroutine print_x

C------- print_c(Xf, Xc, Af, Ac, na, nb) ----------------------

C  Print conveyor status in positions na:nb (for testing)

      subroutine print_c(Xf, Xc, Af, Ac, na, nf)
      INCLUDE 'nbmode.h'
      real, array(1:3, 1:N) :: Xf, Xc, Af, Ac
      integer na, nf

      integer kk
      print 22, (kk, kk = na, nf)
 22   FORMAT ( 10(1X : I9 :) )
      print *, "  - Xf ----------"
      call print_x(Xf, na, nf)
      print *, "  - Xc ----------"
      call print_x(Xc, na, nf)
      print *, "  - Af ----------"
      call print_x(Af, na, nf)
      print *, "  - Ac ----------"
      call print_x(Ac, na, nf)
      end subroutine print_c


C------- Grav(Xf) ------------------------------------

C  Calculate accelerations (Newtonian gravitation)
C  Uses systolic ring calculation, taking advantage
C  of anti-symmetric force.  Data being circulated
C  on 'conveyor' are marked 'c', fixed items are 'f'.

      function Grav(Xf)

      INCLUDE 'nbmode.h'

      real, array(1:3, 1:N) :: Xf, Grav
CMPF  ONDPU Xf
CMF$  LAYOUT Xf(,)
      real, array(1:3, 1:N) :: Af, Ac, D, R, Xc, Mc
```

```
      integer k, lm

      Af = 0.0          ! Fixed accelerations
      Ac = 0.0          ! Circulating accelerations
      Xc = Xf           ! Circulating positions
      Mc = GM           ! Circulating scaled masses
      lm = min(NB, N/2) - 1     ! For loop control
      do, k = 1, lm             ! 2-way accel. steps
        Xc = CSHIFT(Xc, SHIFT=-1, DIM=2)        ! Step conveyor
        Mc = CSHIFT(Mc, SHIFT=-1, DIM=2)
        if ( k > 1 ) then
          Ac = CSHIFT(Ac, SHIFT=-1, DIM=2)
        end if
C  Xc(:,i) = Xf(:,i-k), Similarly Mc and MG
C  Ac(:,i) contributes to particle i-k
        D = Xc - Xf
        R = sqrt(SPREAD(SUM(D*D, 1), 1, 3))
        WHERE ( GM > 0.0 .and. Mc > 0.0 )
          D = D/R**3
          Af = Af + Mc*D
          Ac = Ac - GM*D
        end WHERE
      end do                    ! for 2-way accel.
C  Ac(i) corresponds to Af(i-lm)
      if ( NB > N/2 ) then      ! Final 1-way accel.
        Xc = CSHIFT(Xc, SHIFT=-1, DIM=2)        ! Step conveyor
        Mc = CSHIFT(Mc, SHIFT=-1, DIM=2)
        Ac = CSHIFT(Ac, SHIFT=-1, DIM=2)
        lm = N/2                ! lm gives shift of Ac
        D = Xc - Xf
        R = sqrt(SPREAD(SUM(D*D, 1), 1, 3))
        WHERE ( GM > 0.0 .and. Mc > 0.0 )
          D = D/R**3
          Af = Af + Mc*D
        end WHERE
      end if                    ! for final accel.
D     call print_c(Xf, Xc, Af, Ac, 29, 35)
      Ac = CSHIFT(Ac, SHIFT=lm, DIM=2)    ! Back to correct position
D     call print_c(Xf, Xc, Af, Ac, 1, 7)
C   Combine accels for final result.
      Grav = Af + Ac
      end function Grav


C------- modeul(h, ns)  -------------------------------

C  Perform ns steps of modified Euler with time-step h.
C  X0, V0 are taken to be current state, and are updated
C  with final state.  X1, V1 are used in the process.

      subroutine modeul(h, ns)
      real h
      integer ns

      INCLUDE 'nbmode.h'

      INTERFACE
        function Grav(Xf)
        INCLUDE 'nbmode.h'
        real, array(1:3, 1:N) :: Xf, Grav
CMPF    ONDPU Xf
CMF$    LAYOUT Xf(,)
      END INTERFACE

      real h2
      integer k
```

```
      h2 = 0.5*h          ! h/2, for convenience
      do k = 1, ns        ! ns steps
        X1 = X0 + h2*V0          ! Half-way positions
        V1 = V0 + h2*Grav(X0)    ! Half-way velocities
        X0 = X0 + h*V1           ! Next positions
        V0 = V0 + h*Grav(X1)     ! Next velocities
      end do
      end subroutine modeul


C------- nbmode  ------------------------------

C  Main program:  reads problem and control information
C  from data file named by user, performs and times
C  calculation.  Print control:
C         0 - just performance data
C       < 0 - all
C       > 0 - number given

      program nbmode

      INCLUDE 'nbmode.h'

      integer ns, np, nv, k, PX
      real dt, ar, am
      character*50 fnm

      print '("/ Data file> "$)'     ! Request data file
      read *, fnm
      print '("  Print ctl> "$)'         ! Request print control
      read *, PX
C   Begin initialization
      open(10, file=fnm, status = "OLD")
      read (10, *) NB
      call setup()      ! Setup geometric structure
      call init()       ! Read data for individual bodies
C   Read "asteroid" descriptions from file
      read (10, *) ns
      do while ( ns > 0 )
        read (10, *) ar, am
        call mkast(ns, ar, am)
        read (10, *) ns
      end do
    ' read (10, *) ns, np, dt    ! Integration parameters
      close(10)
      print 21, NB, N
 21   FORMAT (/, X, I4, " bodies, geometry allows ", I4)
      print 22, G, dt
 22   FORMAT ("    G = ", E13.6, "  delta-t = ", E12.5)
      print *, "  simulated time = 0"
      call print_state(PX)
C   Scale masses
      GM = G*m
C   np runs of ns steps, timestep dt
      do, k = 1, np
        call CM_timer_clear(1)
        call CM_timer_start(1)
        call modeul(dt, ns)
        call CM_timer_stop(1)
        call CM_timer_print(1)
        print 23, k*ns*dt
 23     FORMAT (/ " simulated time = ", F9.3)
        call print_state(PX)
      end do
      end program nbmode
```

```
C*****************************************************
C  Data files:
C
C   N                    No. of explicit bodies
C
C   x y z                Position of body 1
C     :                        :
C   x y z                Position of body N
C
C   vx vy va             Velocity of body 1
C      :                     :
C   vx vy va             Velocity of body N
C
C   m                    Mass of body 1
C   :                        :
C   m                    Mass of body N
C
C   G                    Gravitational constant
C
C   nx                   Number,
C   ar am                radius, mass of asteroid group
C      :                        :     0 or more groups
C   nx                   Number,
C   ar am                radius, mass of asteroid group
C   -1                   Stopper
C
C   ns                   Number of time steps per phase
C   np                   Number of phases
C   dt                   delta-t (per time step)
C
C  Further data in the file will be ignored.
C*****************************************************
```

nb2mode.fcm

```
C  N-body calculation using the modified Euler method.
C  Uses O(N^2) processors to calculate accelerations.
C
C  Data arrays (position, velocity, acceleration, mass)
C  are (N, N, 3), organized:
C
C
C     /|  z /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/|
C   (c)  y /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/ |
C   /    x /_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/ /|
C        1 |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|///|
C   |      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   |      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   P      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   a      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   r      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   t      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   i      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   c      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   l      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   e      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C          |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C  (i)     |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//|
C   |      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|///
C   v      |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|//
C        N |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|/
C
C          1   ---  Particle (j)  -->    N
C
C  The parameter N and common block /NBODYPE/
C  with arrays
C
C       X0, X1  - Current, next positions
C       V0, V1  - velocities
C        MI,GJ  - mass(i), G*mass(j)
C        i,j,c  - array coordinates
C         diag  - diagonal entries
C          act  - active in acceleration
C
C  and common block /NBODYFE/ with
C
C           NB  - integer <= N, actual no. of bodies
C            G  - real, the gravitational constant
C
C  are defined in the file nb2mode.h
C
C  System wide conventions regarding data:
C
C       Entries (i, :, :)  - data for particle i
C  except    GJ (:, j, :) = G*mass(j)
C
C  This is intentionally redundant, and the programs
C  perform many redundant calculations.
C
C  File:  nb2mode.fcm


C------- setup() ----------------------------------------

C  Initialize coordinates, logical flags.

      subroutine setup()

      INCLUDE 'nb2mode.h'
      i = SPREAD(SPREAD([1:N], 2, N), 3, 3)
```

```
      j = SPREAD(SPREAD([1:N], 1, N), 3, 3)
      c = SPREAD(SPREAD([1:3], 1, N), 2, N)
      diag = (i == j)
      actv = .FALSE.
      end subroutine setup


C------- init() ----------------------------------------

C  Load initial positions, velocities, also masses,
C  gravitational const.  All are read from logical
C  unit 10.  Replicates masses over x,y,z.
C  Data for particle i are left in (i, :, :) .

      subroutine init()

      INCLUDE 'nb2mode.h'

      integer k
      real mx
      X0 = 0.0            ! For unused particles
      V0 = 0.0
      do, k = 1, NB
        read (10, *) X0(k, 1, 1), X0(k, 1, 2), X0(k, 1, 3)
      end do
      do, k = 1, NB
        read (10, *) V0(k, 1, 1), V0(k, 1, 2), V0(k, 1, 3)
      end do
      M = 0.0             ! For unused particles
      do, k = 1, NB
        read (10, *) mx
        MI(k, 1, :) = mx
      end do
      read (10, *) G
      X0 = SPREAD(X0(:, 1, :), 2, N)
      V0 = SPREAD(V0(:, 1, :), 2, N)
      MI = SPREAD(MI(:, 1, :), 2, N)
      end subroutine init


C------- mkast(nx, ar, am) ----------------------------

C  Insert nx asteroids at radius ar with total mass am.
C  This assumes that an orbit of radius 1 about origin
C  corresponds to speed 2*pi (e.g., au-earth-yr units).
C  Data for particle i are left in  (i, :, :) .

      subroutine mkast(nx, ar, am)
      integer nx          ! Number of new asteroids
      real ar, am         ! Orbit radius, total mass

      INCLUDE 'nb2mode.h'

      real, parameter:: TWOPI = 6.2831853
      real, array(1:N, 1:N, 1:3) :: sx, cx, arg
      real spd
      integer nn
      print 21, nx, ar, am
 21   FORMAT (" Making ", I4, " asteroids, radius ",
     *        F10.5, ", mass ", F8.5)
      nn = NB + nx
      spd = TWOPI/sqrt(ar)
      where ( (NB < i) .and. (i <= nn) )
        arg = (TWOPI*(i - NB))/nx
        sx = sin(arg)
```

```
        cx = cos(arg)
      end where
      where ( (NB < i) .and. (i <= nn) .and. (c == 1) )
        X0 =  cx
        V0 = -sx
      end where
      where ( (NB < i) .and. (i <= nn) .and. (c == 2) )
        X0 =  sx
        V0 =  cx
      end where
      where ( (NB < i) .and. (i <= nn) .and. (c == 3) )
        X0 = 0.0
        V0 = 0.0
      end where
      where ( (NB < i) .and. (i <= nn) )
        X0 = X0*ar
        V0 = V0*spd
        MI = am/nx
      end where
      NB = nn
      end subroutine mkast


C------- print_state(PX) ------------------------------

C  Print current state of system according to PX.
C  See comments for main program for PX settings.

      subroutine print_state(PX)
      integer PX        ! Print control

      INCLUDE 'nb2mode.h'

      integer k, np
      np = NB
      if ( (PX >= 0) .and. (PX < NB) ) np = PX
      print 21
 21   FORMAT (/ "          x          y          z      ",
     *          "    vx         vy         vz         m")
      do, k = 1, np
        print 22, k, X0(k,1,1), X0(k,1,2), X0(k,1,3),
     *               V0(k,1,1), V0(k,1,2), V0(k,1,3), MI(k,1,1)
      end do
 22   FORMAT (1X, I3, 6F10.5, E10.2)
      end subroutine print_state


C------- print_x(ax, nx, cx) ------------------------------

C  Print ax(1:nx, 1:nx, cx), nx <= 10 (for testing)

      subroutine print_x(ax, nx, cx)

      INCLUDE 'nb2mode.h'

      real, array(1:N, 1:N, 1:3) :: ax
CMPF  ONDPU ax
CMF$  LAYOUT ax(,,)
      integer nx, cx

      integer u, lm
      lm = min(nx, 10)
      do, u = 1, lm
        print 21, ax(u, 1:lm, cx)
 21     FORMAT ( 10(1X : E9.2 :) )
      end do
```

```
      end subroutine print_x

C------- Grav(Xf) ------------------------------------

C  Calculate accelerations (Newtonian gravitation)
C  Uses completely parallel calculation, ignoring
C  anti-symmetry of force.  Acceleration for body i is
C  left in  (i, :, :)  of result.
C  Acceleration on body i due to body j is calculated
C  in entries (i, j, :) .

      function Grav(Xf)

      INCLUDE 'nb2mode.h'

      real, array(1:N, 1:N, 1:3) :: Xf, Grav
CMPF  ONDPU Xf
CMF$  LAYOUT Xf(,,)

      real, array(1:N, 1:N, 1:3) :: Ac, D, R, Xj
      integer k, lm

C   Spread Xf across j from diagonal:
      where ( diag )
        Xj = Xf
      elsewhere
        Xj = -1E30                ! Enormously negative
      end where
      Xj = SPREAD(MAXVAL(Xj, 1), 1, N)
D     print *, " --- Xj -----------"
D     call print_x(Xj, 7, 1)      ! For testing
C   Now Xj(i,j,:) = X0(j,j,:)

      Ac = 0.0
      where ( actv )
        D  = Xj - Xf              ! Displacement
      elsewhere
        D = 1.0              ! Protects from div by 0
      endwhere
D     print *, " --- D  -----------"
D     call print_x(D, 7, 1)       ! For testing
      R  = sqrt(SPREAD(SUM(D*D, 3), 3, 3))
D     print *, " --- R  -----------"
D     call print_x(R, 7, 1)       ! For testing
D     print *, " MIN(R) = ", MINVAL(R)
      where ( actv )  Ac = GJ*D/R**3
D     print *, " --- Ac  -----------"
D     call print_x(Ac, 7, 1)      ! For testing
      Grav = SPREAD(SUM(Ac, 2), 2, N)
      end function Grav


C------- modeul(h, ns)  -------------------------------

C  Perform ns steps of modified Euler with time-step h.
C  X0, V0 are taken to be current state, and are updated
C  with final state.  X1, V1 are used in the process.
C  Note that X's, V's are calculated redundantly.

      subroutine modeul(h, ns)
      real h
      integer ns

      INCLUDE 'nb2mode.h'
```

```
      INTERFACE
        function Grav(Xf)
        INCLUDE 'nb2mode.h'
        real, array(1:N, 1:N, 1:3) :: Xf, Grav
CMPF    ONDPU Xf
CMF$    LAYOUT Xf(,,)
      END INTERFACE

      real h2
      integer k

      h2 = 0.5*h          ! h/2, for convenience
      do k = 1, ns        ! ns steps
D     print *, " --- X0 -----------"
D     call print_x(X0, 7, 1)     ! For testing
        X1 = X0 + h2*V0          ! Half-way positions
        V1 = V0 + h2*Grav(X0)    ! Half-way velocities
        X0 = X0 + h*V1           ! Next positions
        V0 = V0 + h*Grav(X1)     ! Next velocities
      end do
      end subroutine modeul


C------- nb2mode  ------------------------------

C  Main program:  reads problem and control information
C  from data file named by user, performs and times
C  calculation.  Print control:
C          0 - just performance data
C        < 0 - all
C        > 0 - number given

      program nb2mode

      INCLUDE 'nb2mode.h'

      integer ns, np, nv, k, PX
      real dt, ar, am
      character*50 fnm

      print '("/ Data file> "$)'     ! Request data file
      read *, fnm
      print '("  Print ctl> "$)'         ! Request print control
      read *, PX
C   Begin initialization
      open(10, file=fnm, status = "OLD")
      read (10, *) NB
      call setup()      ! Setup geometric structure
      call init()       ! Read data for individual bodies
C   Read "asteroid" descriptions from file
      read (10, *) ns
      do while ( ns > 0 )
        read (10, *) ar, am
        call mkast(ns, ar, am)
        read (10, *) ns
      end do
      read (10, *) ns, np, dt    ! Integration parameters
      close(10)
      print 21, NB, N
 21   FORMAT (/, X, I4, " bodies, geometry allows ", I4)
      print 22, G, dt
 22   FORMAT ("    G = ", E13.6, "  delta-t = ", E12.5)
      print *, "  simulated time = 0"
      call print_state(PX)
C   Scale masses and distribute, set actv
      where ( diag )
```

```
        GJ = G*MI
      elsewhere
        GJ = -1E30                  ! Enormously negative
      end where
      GJ = SPREAD(MAXVAL(GJ, 1), 1, N)
D     print *, " --- GJ  -----------"
D     call print_x(GJ, 7, 1)     ! For testing
      actv = (i <= NB) .and. (j <= NB) .and. .not. diag
D     print *, " ---actv------------"
D     do, k = 1, 7
D       print *, actv(k, 1:7, 1)
D     end do
C   np runs of ns steps, timestep dt
      do, k = 1, np
        call CM_timer_clear(1)
        call CM_timer_start(1)
        call modeul(dt, ns)
        call CM_timer_stop(1)
        call CM_timer_print(1)
        print 23, k*ns*dt
 23     FORMAT (/ " simulated time = ", F9.3)
        call print_state(PX)
      end do
      end program nb2mode

C*****************************************************
C  Data files:  See nbmode.f
C*****************************************************
```

```
C  Header file for nb2mode.fcm .  See that file for
C  explanation.

      integer, parameter:: N = 64
      real, array(1:N, 1:N, 1:3) :: X0, X1, V0, V1, MI, GJ
CMPF  ONDPU X0, X1, V0, V1, MI, GJ
CMF$  LAYOUT X0(,,), X1(,,), V0(,,), V1(,,), MI(,,), GJ(,,)
      COMMON /NBODYPE/ X0, X1, V0, V1, MI, GJ
      integer, array(1:N, 1:N, 1:3) :: c, i, j
CMPF  ONDPU c, i, j
CMF$  LAYOUT c(,,), i(,,), j(,,)
      COMMON /NBODYPE/ c, i, j
      logical, array(1:N, 1:N, 1:3) :: diag, actv
CMPF  ONDPU diag, actv
CMF$  LAYOUT diag(,,), actv(,,)
      COMMON /NBODYPE/ diag, actv
      integer NB
      COMMON /NBODYFE/ NB
      real G
      COMMON /NBODYFE/ G
```

```
C  Header file for nbmode.fcm .  See that file for
C  explanation.

      integer, parameter:: N = 64
      real, array(1:3, 1:N) :: X0, X1, V0, V1, M, GM
CMPF  ONDPU X0, X1, V0, V1, M, GM
CMF$  LAYOUT X0(,), X1(,), V0(,), V1(,), M(,), GM(,)
      COMMON /NBODYPE/ X0, X1, V0, V1, M, GM
      integer, array(1:3, 1:N) :: c, i
CMPF  ONDPU c, i
CMF$  LAYOUT c(,), i(,)
      COMMON /NBODYPE/ c, i
      integer NB
      COMMON /NBODYFE/ NB
      real G
      COMMON /NBODYFE/ G
```

```
 7

 -0.2594970  -0.9809988   0.0000000
  0.9996320   1.1499410   0.0000000
  1.4145586   4.8663114  -0.0515851
 -6.8029321   6.1397428   0.1652675
-14.1847610 -11.9941410   0.1396083
 -7.9557686 -29.2149712   0.7840482
  0.0000000   0.0000000   0.0000000

  5.9726963  -1.6298815   0.0000000
 -3.2802815   3.8956787   0.1898075
 -2.6809127   0.8989233  -0.0565152
 -1.4776006  -1.5171660   0.0851270
  0.9138937  -1.1634981  -0.0161611
  1.0961944  -0.2943595  -0.0193314
  0.0000000   0.0000000   0.0000000

     1.00
     0.11
   317.84
    95.17
    14.54
    17.25
332958.00

0.0001185684121

9
2.8 0.01
-1

20 5 0.01 0.000001

-8.0 8.0 -4.5 4.5
EMJSUNO
```

```
 7

 -0.2594970  -0.9809988   0.0000000
  0.9996320   1.1499410   0.0000000
  1.4145586   4.8663114  -0.0515851
 -6.8029321   6.1397428   0.1652675
-14.1847610 -11.9941410   0.1396083
 -7.9557686 -29.2149712   0.7840482
  0.0000000   0.0000000   0.0000000

  5.9726963  -1.6298815   0.0000000
 -3.2802815   3.8956787   0.1898075
 -2.6809127   0.8989233  -0.0565152
 -1.4776006  -1.5171660   0.0851270
  0.9138937  -1.1634981  -0.0161611
  1.0961944  -0.2943595  -0.0193314
  0.0000000   0.0000000   0.0000000

     1.00
     0.11
   317.84
    95.17
    14.54
    17.25
332958.00

0.0001185684121

 9
2.8 0.01
16
2.5 0.01
-1

20 5 0.01 0.000001

-8.0 8.0 -4.5 4.5
EMJSUNO
```

```
 7

 -0.2594970  -0.9809988   0.0000000
  0.9996320   1.1499410   0.0000000
  1.4145586   4.8663114  -0.0515851
 -6.8029321   6.1397428   0.1652675
-14.1847610 -11.9941410   0.1396083
 -7.9557686 -29.2149712   0.7840482
  0.0000000   0.0000000   0.0000000

  5.9726963  -1.6298815   0.0000000
 -3.2802815   3.8956787   0.1898075
 -2.6809127   0.8989233  -0.0565152
 -1.4776006  -1.5171660   0.0851270
  0.9138937  -1.1634981  -0.0161611
  1.0961944  -0.2943595  -0.0193314
  0.0000000   0.0000000   0.0000000

      1.00
      0.11
    317.84
     95.17
     14.54
     17.25
332958.00

0.0001185684121

 9
2.8 0.01
16
2.5 0.01
32
2.7 0.01
-1

20 5 0.01 0.000001

-8.0 8.0 -4.5 4.5
EMJSUNO
```

```
 7

 -0.2594970  -0.9809988   0.0000000
  0.9996320   1.1499410   0.0000000
  1.4145586   4.8663114  -0.0515851
 -6.8029321   6.1397428   0.1652675
-14.1847610 -11.9941410   0.1396083
 -7.9557686 -29.2149712   0.7840482
  0.0000000   0.0000000   0.0000000

  5.9726963  -1.6298815   0.0000000
 -3.2802815   3.8956787   0.1898075
 -2.6809127   0.8989233  -0.0565152
 -1.4776006  -1.5171660   0.0851270
  0.9138937  -1.1634981  -0.0161611
  1.0961944  -0.2943595  -0.0193314
  0.0000000   0.0000000   0.0000000

     1.00
     0.11
   317.84
    95.17
    14.54
    17.25
332958.00

0.0001185684121

 9
2.8 0.01
16
2.5 0.01
32
2.7 0.01
-1

200 5 0.001
```

```
 7

 -0.2594970  -0.9809988   0.0000000
  0.9996320   1.1499410   0.0000000
  1.4145586   4.8663114  -0.0515851
 -6.8029321   6.1397428   0.1652675
-14.1847610 -11.9941410   0.1396083
 -7.9557686 -29.2149712   0.7840482
  0.0000000   0.0000000   0.0000000

  5.9726963  -1.6298815   0.0000000
 -3.2802815   3.8956787   0.1898075
 -2.6809127   0.8989233  -0.0565152
 -1.4776006  -1.5171660   0.0851270
  0.9138937  -1.1634981  -0.0161611
  1.0961944  -0.2943595  -0.0193314
  0.0000000   0.0000000   0.0000000

     1.00
     0.11
   317.84
    95.17
    14.54
    17.25
332958.00

0.0001185684121

1
2.8 0.01
-1

20 5 0.01 0.000001

-8.0 8.0 -4.5 4.5
EMJSUNO
```

pde.notes

Some Essentials of Partial Differential Equation's

These notes incorporate much material from notes by G.C. Fox for CPS 615, Fall 1992.

Three classical, 2nd-order PDE's

Following earlier conventions, for a function U(x, y, ...) we write

$$Uxx = \frac{d^2 U}{dx^2} \quad \text{(the second partial derivative)}$$

and define the "Laplacian" of U

Lap(U) = Uxx + Uyy in 2 (space) dimensions

Lap(U) = Uxx + Uyy + Uzz in 3 (space) dimensions

and so on for higher dimensions.

Alternate notation:

$$Lap(U) = \nabla^2 U \qquad [\ \text{sometimes } \Delta U\ ]$$

Laplace's Equation:

Lap(U) = 0 [elliptic]

Inhomogeneous form:

Lap(U) = c(x,y) [Poisson's equation]

Wave Equation:

$$Lap(U) = \frac{1}{C^2} * Utt \qquad [\ \text{hyperbolic}\ ]$$

Here U is a function of t in addtion to x, y, ...

Diffusion Equation (or heat equation):

$$Lap(U) = \frac{1}{K} * Ut \qquad [\ \text{parabolic}\ ]$$

Boundary Conditions

Consider a 2nd-order ORDINARY DE:

a(x)Uxx + b(x)*Ux + c(x)*U = d(x)

where a, b, c, d are smooth functions.

Wherever a(x) != 0, if we know U(x) and Ux(x) then

The equation gives Uxx(x)

Repeated differentiation of the equation yields all higher derivatives

so

U is completely determined in the neighborhood of x by U(x), U'(x).

Boundary conditions for PDE's in 2 space dimensions:

In analogy to specifying U(x), U'(x) for ODE, it is natural to specify boundary conditions along a "curve" C

```
      /                 /
     /                 / C
    /                 /
    |--> n           /
    |               /
    \______________/
```

Cauchy boundary data:

U, dU/dn (normal derivative) specified on C

These already determine dU/ds (derivative tangent to C), so they could plausibly be sufficient to determine the solution.

Boundary data on a closed curve:

```
      _____________\
     /             | C
    /              |
    |--> - - - - <--|    Giving U and dU/dn at both ends
    |               |    looks like overspecification.
    \_____________/
```

Dirichlet boundary data:

U specified at every point of C

Neumann boundary data:

dU/dn specified at every point of C

The appropriate boundary data depends on the nature of the equation.

Characteristics

Consider a general 2nd-order equation

a*Uxx + b*Uxy + c*Uyy = E

where E may involve U, Ux, Uy, x, and y.

Cauchy data at a particular boundary point (x0, y0) in effect give

U(x0, y0), p = Ux(x0, y0), q = Uy(x0, y0)

These give E(x0, y0) directly.

Write

$$r = Uxx(x0, y0), \quad s = Uxy(x0, y0), \quad t = Uyy(x0, y0)$$

If (x0+dx, y0+dy) is an infinitesimal displacement ALONG C we have

$$dp = r*dx + s*dy$$

$$dq = \qquad s*dx + t*dy$$

$$E = r*a + s*b + t*c$$

r, s, t are uniquely determined if the determinant

$$\begin{vmatrix} dx & dy & 0 \\ 0 & dx & dy \\ a & b & c \end{vmatrix} \neq 0$$

Put another way, if

$$a*(dy)^2 + b*dx*dy + c*(dx)^2 = 0$$

or [Characteristic equation]

$$a*(dy/dx)^2 + b*(dy/dx) + c = 0$$

then Cauchy data on C do not determine the solution at (x0, y0).

At any point (x0, y0), the 'characteristic equation' has 0, 1, or 2 solutions defining the slope(s) dy/dx of the 'characteristic curves' through (x0, y0). There are three cases:

$b^2 > 4*a*c$

two real characteristics - a hyperbolic equation

Wave equation: $Uxx = \frac{1}{C^2}*Utt$

$a = 1, \quad b = 0, \quad c = -1/C^2$

$(dt/dx)^2 = 1/C^2$ or $dx/dt = +/- C$

Characteristics: $x +/- C*t = \text{constant}$

$b^2 = 4*a*c$

one real characteristic - a parabolic equation

Diffusion equation: $Uxx = \frac{1}{K}*Ut$

$a = 1, \quad b = c = 0$

$(dt/dx)^2 = 0$ or $dt/dx = 0$

Characteristics: $t = \text{constant}$

$b^2 < 4*a*c$

no real characteristics - an elliptic equation

Laplace's equation: $Uxx + Uyy = 0$

$a = 1, \quad b = 0, \quad c = 1$

$(dy/dx)^2 = 0$ - No solutions

When a, b, c actually depend on x, y, t the equation can change character from one region to another.

Usual boundary conditions

Equation	Boundary Curve	Boundary Data
Hyperbolic	Open	Cauchy
Elliptic	Closed	Dirichlet/Neumann
Parabolic	Open	Dirichlet/Neumann

Hyperbolic Boundary Conditions

We may assume $C = 1$, $U_{xx} - U_{tt} = 0$

```
t
^
|            \     \     \     \/     /     /     /
|             \     \     \    /\    /     /     /
|   x + t = w  \     \     \  /  \  /     /     / x - t = v
|               \     \     \/    \/     /     /
|                \     \    /\D   /\    /     /
|                 \     \  /--\  /  \  /     /
|                  \     \/----\/    \/     /
|                   \    /\----/\    /\    /
|                    \  /--\--/--\  /  \  /
|___________________A\/====\/====\/B___\/_______> x
                     /\====/\====/\    /\
                    /  \--/--\--/  \  /  \
                   /    \/----\/    \/    \
                  /     /\----/\    /\     \
                 /     /  \--/  \  /  \     \
                /     /    \/C   \/    \     \
                          '
```

Change coordinates to v, w using $x + t = w$, $x - t = v$, or

$x = (v + w)/2, \quad t = (w - v)/2$

It is easy to see that the wave equation becomes

$$U_{vw} = 0$$

which is solved by

$$U = f(v) + g(w), \quad f \text{ and } v \text{ any differentiable functions!}$$

Suppose we have Cauchy data on A--B, namely $U(x, 0)$ and $U_t(x, 0)$.

Then

$$\left.\begin{aligned} U(x, 0) &= f(x) + g(x) \\ U_t(x, 0) &= -f'(x) + g'(x) \end{aligned}\right. \quad \text{on A--B}$$

We can obtain a solution by assuming that g(A) = f(A), so

$$g(x) - f(x) = \text{INTEGRAL}_{r=A}^{x}\ g'(r) - f'(r)\ dr = \text{INTEGRAL}_{r=A}^{x}\ Ut(r,0)\ dr$$

whence

$$f(x) = 0.5*[\ U(x,\ 0) - (g(x) - f(x))\ ]$$

$$g(x) = 0.5*[\ U(x,\ 0) + (g(x) - f(x))\ ]$$

From the figure we see that the solution is determined uniquely in the diamond ACBD but not outside it.

More complicated situations can also be analyzed, but in general the solution is determined only in a bounded region limited by the extent or the data.

Elliptic Boundary Conditions

We have already discussed Laplace's Equation. Dirichlet data clearly determine a unique solution:

Suppose

- U1, U2 are solutions of Laplace's equation in a region R
- U1 = U2 on the boundary of R (a closed curve)

then

- U1 - U2 is a solution which is 0 on the boundary
- by the maximum principle, U1 - U2 = 0 in the interior.

One can also show that Neumann data determine the solution up to an additive constant. Cauchy data clearly overdetermine the problem.

Parabolic Boundary Conditions

For the diffusion equation, Uxx = Ut, the characteristics are the lines t = constant, e.g., t = 0.

Reasonable data:

```
                                  t
                                  ^
                                  |
                                  |
                                  |
                                  |
      . . . ______________________|______________________ . . .  x
```

U(x, 0) or Ut(x, 0) for all x (not both)

Determines solution for t > 0

Example:

```
                  |¯ 0 for x < 0
```

```
U(x, 0)  =  |
            |_ 1 for x > 0
```

One easily checks that the solution is

```
                 1          x/sqrt(t)
U(x, t) = ---------- *  INTEGRAL exp(-v^2/4) dv
          2*sqrt(pi)     v = -inf
```

This can be derived from the scale invariance of the problem, from which one sees that

```
U(x, t) = f(w) for some function f,  where w = x/sqrt(t) .
```

```
                 U(x, 0)                          U(x, t), t > 0
      |       *                              *
      |       *                              *
      |       *                              *
      |       *                              *
      |       *                              *
      |       *                          ---  *
      |       *                           ^    *
      |       *                           |     *
  |   |       *                           |       *
x |  __|__     *______________*        O(sqrt(t))        *
  |   |                      *           |                 *
  v   |                      *           |                   *
      |                      *           v                     *
      |                      *          ---                     *
      |                      *                                  *
      |                      *                                  *
      |                      *                                  *
      |                      *                                  *
      |                      *                                  *
```

Another reasonable problem:

```
      t1 |.............................|
         |.............................|
         |.............................|
    ^    |.............................|
    | t  |.............................|
    |    |.............................|
         |.............................|
         |.............................|
         |.............................|
         |.............................|
      0  ---------------------------------
         a              ---->            b
                          x
```

U(x, 0) for a <= x <= b with

U(a, t) and U(b, t) for 0 <= t <= t1
determines solution for a <= x <= b, 0 <= t <= t1

One could instead use Ut(x, 0), Ux(a, t), Ux(b, t)
or some mixture of values and derivatives.

iter.notes

Iterative Methods for Sparse Systems of Linear Equations

Introduction

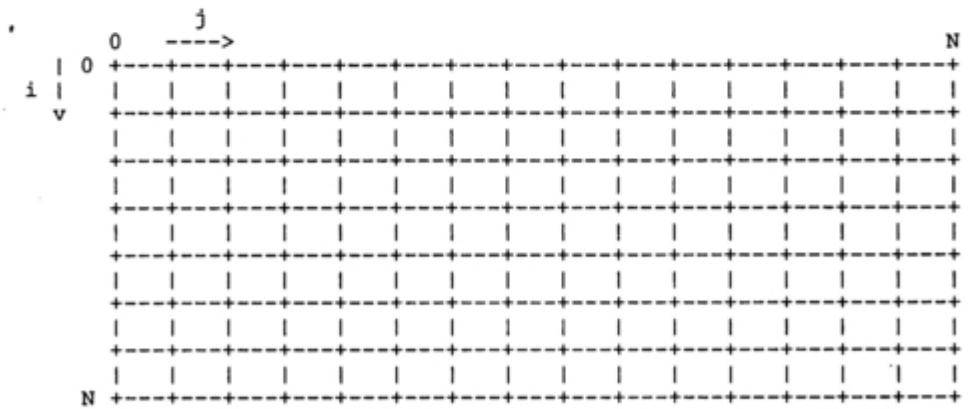
- Useful for solution of sparse systems such as those which arise in the solution of PDE's by finite-difference methods
- Point-Jacobi iteration for Laplace's equation in a rectangle is a classic example (discussed earlier)
- Require storage only proportional to N for a problem with N grid points, rather than O(N^2) for typical direct methods
- Many are amenable to efficient parallel implementation, at least for larger problems
- May suffer from slow convergence, especially with bad starting guesses
- Each iteration suppresses rounding error from previous iterations, so methods are relatively insensitive to rounding error

Basic Iterative Methods

We illustrate using the discrete Laplace equation in a rectangle with the familiar 5-point operator

$$U(i, j) = \{ U(i+1, j) + U(i-1, j) + U(i, j+1) + U(i, j-1) \}/4$$

to be satisfied in the interior of U(0:M, 0:N).



Point-Jacobi:

$$U(1:M-1, 1:N-1) := 0.25*(U(0:M-2, 1:N-1) + U(2:M, 1:N-1) + U(1:M-1, 0:N-2) + U(1:M-1, 2:N))$$

- Easy and efficient to implement in parallel
- Rather slow convergence

Gauss-Seidel:

- Many found the Jacobi iteration awkward in a sequential implementation - so much so that several implemented Gauss-Seidel inadvertantly.

Natural Sequential version:

```
do i = 1, M-1
  do j = 1, N-1
    U(i,j) := 0.25*( U(i+1, j) + U(i-1, j) + U(i, j+1) + U(i, j-1) )
  end do ! j
end do ! i
```

- Each step uses most recently calculated values

- Gives better convergence than Jacobi with same effort

- Terrible for parallel calculation as stated

  BUT: order of updates need not be the order just given!

Red/black ordering for Gauss-Seidel:

```
            j
      0   ---->                                                    N
  | 0 R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R
i |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
  v   B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
      R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
      B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
      R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
      B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
      R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R
      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
    M B---R---B---R---B---R---B---R---B---R---B---R---B---R---B---R---B
```

- Neighbors of "Red" points are all "Black"

- Neighbors of "Black points are all "Red"

- A correct version of Gauss-Seidel is to update all red points simultaneously, then update all black points using the red values just calculated

- Implementation discussed later

- Red/Black ordering for 3-D grid:

```
            B---R---B---R---B---R---B---R
           /   /   /   /   /   /   /   /|
      /|   R---B---R---B---R---B---R---B B
  k /     /   /   /   /   /   /   /   /|/|
   /     B---R---B---R---B---R---B---R R R
        /   /   /   /   /   /   /   /|/|/|
  | 0 R---B---R---B---R---B---R---B---R B B B
i |   |   |   |   |   |   |   |   |/|/|/|/|
  v   B---R---B---R---B---R---B---R---B R R R
      |   |   |   |   |   |   |   |   |/|/|/|
```

```
      R---B---R---B---R---B---R---B B B B
      |   |   |   |   |   |   |   |/|/|/
      B---R---B---R---B---R---B---R R R
      |   |   |   |   |   |   |   |/|/
      R---B---R---B---R---B---R---B B
      |   |   |   |   |   |   |   |/
    M B---R---B---R---B---R---B---R
      0   ---->                   N
            j
```

Succesive Over-Relaxation (SOR)

- Gauss-Seidel, though better than Jacobi, is usually too cautious

- A little extrapolation (over-relaxation) often accelerates convergence.

Sequential SOR with relaxation parameter W:

```
do i = 1, M-1
  do j = 1, N-1
    ! Calculate Gauss-Seidel update for U(i, j)
    X:= 0.25*( U(i+1, j) + U(i-1, j) + U(i, j+1) + U(i, j-1) )
    ! Over-relax to obtain SOR update
    U(i,j)  := W*x + (1.0 - W)*U(i, j)
  end do ! j
end do ! i
```

- We take 1.0 < W < 2.0, the precise value determined by numerical experiments (more on this later).

- W = 1.0 gives Gauss-Seidel, W < 1.0 is pointless, W >= 2.0 destroys convergence.

- See redblack.f(cm) for implementation.

cyl. notes

Laplacian in Cylindrical Coordinates

Notation

U is the function in question,

Coordinates: X, Y, R, A (Z is of no concern)

X = R*cos(A), Y = R*sin(A)

Derivatives: Ux, Ur, Ua, Xr, Xa, Uxx, ...

The derivation

```
Ur = Ux*Xr + Uy*Yr

Urr = (Ur)x*Xr + (Ur)y*Yr
    = (Ux*Xr + Uy*Yr)x*Xr + (Ux*Xr + Uy*Yr)y*Yr
    =   (Uxx*Xr + Ux*Xrx + Uyx*Yr + Uy*Yrx)*Xr
      + (Uxy*Xr + Ux*Xry + Uyy*Yr + Uy*Yry)*Yr

Ua = Ux*Xa + Uy*Ya

Uaa = (Ua)x*Xa + (Ua)y*Ya
    = (Ux*Xa + Uy*Ya)x*Xa + (Ux*Xa + Uy*Ya)y*Ya
    =   (Uxx*Xa + Ux*Xax + Uyx*Ya + Uy*Yax)*Xa
      + (Uxy*Xa + Ux*Xay + Uyy*Ya + Uy*Yay)*Ya
```

Now

```
Xr = cos(A)                Yr = sin(A)
Xrx = Xry = Yrx = Yry = 0
Xa = -R*sin(A) = -Y        Ya = R*cos(A) = X
Xay = -1                   Yax = 1
Xax = Yay = 0
```

so

```
Ur  = Ux*cos(A) + Uy*sin(A)

Urr = (Uxx*Xr + Uyx*Yr)*Xr + (Uxy*Xr + Uyy*Yr)*Yr
    = Uxx*Xr^2 + 2*Uxy*Xr*Yr + Uyy*Yr^2
    = Uxx*(cos(A))^2 + 2*Uxy*cos(A)*sin(A) + Uyy*(sin(A))^2

Uaa = (Uxx*Xa + Uyx*Ya + Uy*Yax)*Xa + (Uxy*Xa + Ux*Xay + Uyy*Ya)*Ya
```

$$= Uxx*Xa^2 + 2*Uxy*Xa*Ya + Uyy*Ya^2 + Uy*Yax*Xa + Ux*Xay*Ya$$

$$= Uxx*R^2*(\sin(A))^2 - 2*Uxy*R^2*\sin(A)*\cos(A) + Uyy*R^2*(\cos(A))^2$$

$$- Ux*R*\cos(A) - Uy*R*\sin(A)$$

Clearly now,

$$Ur/R + Urr + Uaa/R^2 = Uxx + Uyy$$

lens.notes

Implementing the Electrostatic Lens Claculation

The grid, with boundary conditions:

```
                U linear
          Nx+------------+
            |            |
            |            |       U = potential
            |            |
        U=0 |     R1     |U=1000
  ^         |            |
  |         |            |      U=1000
  k|   N    | - - - - -  +-----------------+
  |         |            |                 |
            |            |                 |
        U=0 |     R2     |       R3        |U=1000
            |            |                 |
            |            |                 |
          0 +-------------------------------+
            0    Axis     M     Axis       Mx
                       i ---->
```

U linear:

$$U = (1000*i)/M$$

Plausible initialization:

$$U = (1000*i)/Mx \text{ on axis}$$

Linear interpolation in interior

This seems rather better than that given in chapter 9.

Problem size for parallel computation:

Each of R1 union R2, R2 union R3 big enough to fill computer, so M*Nx, N*Mx >= 8K, say.

We can allow smaller problems for convenient testing.

Decomposing the region for data-parallel comp:

Axis calculation is a nuisance, though fortunately fairly simple. To avoid doing it more than once, we consider

Region A = R2 union R3 Region B = R1

What happens at the interface:

Closer view of grid near plate-tube corner:

```
                                  # = Boundary
        *   *   *   *   # |       a = update in A
          Array  B                b = update in B
        *   *   *   *   # |
      - - - - - - - - - - - - - - - - - - -
  N     b   b   b   b   # | #   #   #   #
                      Array A
```

```
N-1   a   a   a   a   a | *   *   *   *
     -------------------+
       *   *   *   *   *   *   *   *   *
                       M
```

Principal arrays correspond logically to:

```
UA(0:Mx, 0:N)      updates  (1:Mx-1, 1:N-1), (1:Mx-1, 0)
     i    k                     i       k         i     k

UB(N-1:Nx, 0:M)    updates  (N:NX-1, 1:M-1)
     k      i                   k       i
```

Why is UB "transposed"? M is likely to be small; DECmpp works best if axis 1 is 128, axis 2 is 64 (or multiples thereof). CM5 is more flexible, but the transposition shouldn't hurt particularly.

Overlap: UB(N-1:N, 0:M) = UA(0:M, N-1:N)

Program arrays:

?A(0:Ax, 0:Rx) using ?A(0:Mx, 0:N)

corresponding to i = 0:Mx by k = 0:N

?B(0:Bx, 0:Dx) using ?B(0:Nx-N+1, 0:M)

corresponding to k = N-1:Nx by i = 0:M

```
      Bx+----------------+        Ax+-----------------+
        |            ^    |          |            ^    |
Nx-N+1  |-------+  k|     |          |           i|    |
        |       |   +->   |        Mx|--------+   +->  |
B:      |       |    i    |          |        |    k   |
        |       |         |          |        |        |
        |       |         |          |        |        |
        |       |         |          |        |        |
        |||||||||         |          |        |        |
      0 +----------------+        M |       =|        |
        0       M       Dx          |       =|        |
              ^                     |       =|        |
      axis 1  |                     |       =|        |
              |                   0 +-----------------+
              ------->              0        N       Rx
              axis 2
```

Constraints on problem parameters:

```
0 < M <= Dx          M < Mx <= Ax

0 < N <= Rx          0 < Nx - N + 1 <= Bx
```

These preclude some trivial cases of no interest.

Block SOR in the red/black ordering

Classification of points

Red/black in the interior of region A

Red/black on the axis in region A (Region "X")

Red/black in the interior of region B
(includes points k = N, 0 < i < M)

Rather than update say A red, B red, then A black, B black, it is easier and quite proper to update

A red
A black
X red
X black
B red
B black

treating it as six "colors" rather than two. In any case, we have to watch out for consistency in the "overlap" between A and B.

Without trying to develop the theory, we should still pick W so that

1 <= W < 2 .

W = 1 corresponds to "block Gauss-Seidel".

Computational effort, performance

For the programs len.fcm, lensor.fcm. lensaux.fcm:

A interior: (N - 1)*(Mx - 1) points

B interior: (M - 1)*(NX - N - 1) points

X interior: (Mx -1) points

Normal update: 8 flops/point

Axis update: 6 flops/point

Arithmetic effort per update cycle:

(8*(N-1) + 6)*(Mx-1) + 8*(M-1)*(Nx - N - 1)

= (8*N - 2)*(Mx-1) + 8*(M-1)*(Nx - N - 1)

Initialization for each group of iterations:

3*{(Ax + 1)*(Rx + 1) + (Bx + 1)*(Dx + 1)}

Maximum problem configuration:

M = Dx, Mx = Ax, N = Rx, Nx = Bx + Rx - 1

Tests so far use Ax = Bx = 127, Dx = Rx = 63, so array dimensions are 128 by 64.

lens.fcm

```
C  Electrostatic lens calculation, based on chapter 7.
C  Uses block SOR with red/black orderings.
C  Auxilliary routines in lensaux.f
C  SOR function in lensor.f
C
C  File:  lens.f          Included file:  lens.h
C
C  Domain is represented by two groups of arrays
C  corresponding to:
C
C           U linear
C      Nx+------------+
C        |            |
C        |            |       U = potential
C        |            |
C     U=0|     B      |U=1000
C   ^    |            |
C   |    |            |      U=1000
C  k|  N | - - - - -  +-----------------+
C   |    |            |                 |
C        |                              |
C     U=0|              A               |U=1000
C        |                              |
C        |            |                 |
C      0 +------------------------------+
C        0    Axis    M     Axis        Mx
C                   i ---->
C
C
C  Parameters and global entities are defined in
C  the file  lens.h  using common blocks
C
C  /LENSFE/ - scalars       /LENSPE/ - arrays
C
C  Ax, Bx, Dx, Rx - Parameters for array dim's
C
C   M, Mx,  N, NX - Problem geometry
C  IA, KA, IB, KB - Problem coordinate arrays
C          UA, UB - The potential
C     Ared, Bred  - Mark red interior entries
C     Ablk, Bblk  - Mark black interior entries
C     Xred, Xblk  - Mark red/black on axis
C       '     W   - SOR parameter

C  Manage tests and examination of result.

      program testlens
      INCLUDE 'lens.h'

      INTERFACE
        logical function checklens()
      END INTERFACE
C      INTERFACE        ! Compiler doesn't seem to get it
C        real function SORlens(it)
C        integer it
C      END INTERFACE

      character*50 buff      ! For terminal input
      integer iter, titer
      real Up, eps, err
      double precision eltm     ! For elapsed time on CM
      logical more

      more = .TRUE.
      do while ( more )
        print  '(" M, Mx, N, Nx > " $)'
```

```
        read *, M, Mx, N, Nx
        more = .not. checklens()
      end do
      call setuplens()

      print '("potential, iterations, epsilon, W > " $)'
      read *, Up, iter, eps, W
      call initlens(Up)
      err = 1.0 + 2.0*eps   ! To assure iterations
      titer = 0
      do while ( err > eps )
        call CM_timer_clear(1)
        call CM_timer_start(1)
        err = SORlens(iter)
        call CM_timer_stop(1)
        call CM_timer_print(1)
        titer = titer + iter
        print 21, titer, err
 21     FORMAT (1X, I4, " iterations,  Err = ", E13.6)
      end do
C  Browse results until user decides to quit:
      more = .TRUE.
      print *, "Specify ranges in problem coordinates."
      do while ( more )
        print '(" Browse? ( A | B | N | n ) > " $)'
        read *, buff
        if ( (buff(1:1) == 'N') .or. (buff(1:1) == 'n') )
     *      more = .FALSE.
        if ( (buff(1:1) == 'A')
     *      .or. (buff(1:1) == 'B')) then
          print '( " i1, i2, k1, k2 > " $)'
          read *, i1, i2, k1, k2
        end if
        if ( buff(1:1) == 'A' )
     *    call prA(i1, i2, k1, k2)
        if ( buff(1:1) == 'B' )
     *     call prb(i1, i2, k1, k2)
      end do
      end program testlens
```

lens aux.tcm

```
C  Electrostatic lens calculation, based on chapter 7.
C  Initialization and output routines
C  See  lens.f  for further notes.
C
C  File:  lens.faux          Included file:  lens.h


C  Check problem parameters for consistency and
C  compatibility with array dimensions.  Reports
C  problems to standard output.  Returns TRUE
C  if parameters OK, FALSE otherwise.

      logical function checklens()
      INCLUDE 'lens.h'
      integer nd
      logical OK
      OK = .TRUE.
      nd = Nx - N + 1
      if ( .not. (0 < M .and. M <= Dx) ) then
        print *, "M must satisfy 0 < M <= ", Dx
        OK = .FALSE.
      end if
      if ( .not. (M < Mx .and. M <= Ax) ) then
        print *, "M and Mx must satisfy"
        print *, " M < Mx <= ", Ax
        OK = .FALSE.
      end if
      if ( .not. (0 < N .and. N <= Rx) ) then
        print *, "N must satisfy 0 < N <= ", Rx
        OK = .FALSE.
      end if
      if ( .not. (1 < nd .and. nd <= Bx) ) then
        print *, "We require N < Nx and"
        print *, "Nx - N <= ", Bx - 1
        OK = .FALSE.
      end if
      checklens = OK
      end function checklens

C  Setup problem geometry - fills IA, KA, IB, KB
C  including positions not used in current problem.
C  Sets red/black flags for updated entries.
C  Uses problem parameters from /LENSFE/

      subroutine setuplens()
      INCLUDE 'lens.h'
      IA = SPREAD([0:Ax], 2, Rx+1)
      KA = SPREAD([0:Rx], 1, Ax+1)
      Ared = (0 < IA) .and. (IA < Mx) .and.
     *       (0 < KA) .and. (KA <  N)
      Ablk = Ared                          ! Interior
      Ared = Ared .and. (Iand(IA - KA, 1) == 0)
      Ablk = Ablk .and. (.not. Ared)
      Xred = (KA == 0) .and. (0 < IA) .and. (IA < MX)
      Xblk = Xred                        ! Axis interior
      Xred = Xred .and. (Iand(IA - KA, 1) == 0)
      Xblk = Xblk .and. (.not. Xred)
      IB = SPREAD([0:Dx], 1, Bx+1)
      KB = SPREAD([0:Bx], 2, Dx+1) + N-1
      Bred = (0 <  IB) .and. (IB <  M) .and.
     *       (N <= KB) .and. (KB < Nx)
      Bblk = Bred                          ! Interior
      Bred = Bred .and. (Iand(IB - KB, 1) == 0)
      Bblk = Bblk .and. (.not. Bred)
      end subroutine setuplens
```

```
C  Initialize boundary values and interior for boundary
C  potential p (typically 1000.0).  Uses problem
C  parameters and coordinate arrays from /LENS??/.
C  The update sequence is presumed to start with A.

      subroutine initlens(p)
      real p, nmx, fnm, fn, fmx
      INCLUDE 'lens.h'
      nmx = N*Mx          ! For interpolation
      fnm = N*M
      fn = N
      UA = 0.0            ! Unused entries and mid-plane
      where ( IA == Mx ) UA = p          ! Right end
      where ( KA ==  0 .and. 0 < IA .and. IA < Mx )
     *  UA = (IA*p)/Mx            ! Axis
      where ( KA == N .and. M <= IA .and. IA < MX )
     *  UA = p                    ! Tube
      where ( KA == N .and. 0 < IA .and. IA < M )
     *  UA = (IA*p)/M
      where (       0 < KA .and. KA < N     ! Left
     *        .and. 0 < IA .and. IA < M )   ! interior
     *  UA = IA*p*(KA/fnm + (N - KA)/nmx)
      where (       0 < KA .and. KA < N     ! Right
     *        .and. M <= IA .and. IA < Mx ) ! interior
     *  UA = p*(KA/fn + IA*(N - KA)/nmx)
      UB = 0.0   ! Note that K = N-1 isn't important
      where ( IB == M ) UB = p      ! Plate
      where ( 0 < IB .and. IB < M .and. KB <= Nx)
     *  UB = (IB*p)/M     ! Interior and top
      end subroutine initlens


C  Routines for printing UA, UB adjust coordinate
C  ranges so that they fall in arrays and so that
C  output is at most 80 characters/line, but they
C  may include entries in unused portions of arrays.
C  Parameters are problem coordinates, not array
C  coordinates.

C  Print UA(i1:i2, k1:k2) using F8.2 format.

      subroutine prA(i1, i2, k1, k2)
      integer i1, i2, k1, k2
      INCLUDE 'lens.h'
      integer kk
      print *, "UA:"
      if ( i2 > Ax ) i2 = Ax
      if ( i1 <  0 ) i1 =  0
      if ( i2 < i1 ) i2 = i1
      if ( i2 - i1 > 8 ) i2 = i1 + 8
      if ( k2 > Rx ) k2 = Rx
      if ( k1 <  0 ) k1 =  0
      if ( k2 < k1 ) k2 = k1
      print 21, [i1:i2]
 21   FORMAT (" k\i ", 9(I8:))
      do, kk = k2, k1, -1
        print 22, kk, UA(i1:i2, kk)
 22     FORMAT (1X, I4, 9(F8.2:))
      end do
      end subroutine prA

C  Print UB(i1:i2, k1:k2) using F8.2 format.

      subroutine prB(i1, i2, k1, k2)
      integer i1, i2, k1, k2
      INCLUDE 'lens.h'
```

```
      integer kk
      if ( i2 > Dx ) i2 = Dx
      if ( i1 <  0 ) i2 =  0
      if ( i2 < i1 ) i2 = i1
      if ( i2 - i1 > 8 ) i2 = i1 + 8
      if ( k2-N+1 > Bx ) k2 = Bx + N - 1
      if ( k1 < N-1 ) k1 = N - 1
      if ( k2 < k1 ) k2 = k1
      print *, "UB:"
      print 21, [i1:i2]
 21   FORMAT (" k\i ", 9(I8:))
      do, kk = k2, k1, -1
        print 22, kk, UB(kk-N+1, i1:i2)
 22     FORMAT (1X, I4, 9(F8.2:))
      end do
      end subroutine prB

Cnnnn*SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

lensor.fcm

```
C  Electrostatic lens calculation, based on chapter 7.
C  SOR function for  lens.fcm
C  Auxilliary routines in lensaux.fcm
C
C  File:  lensor.fcm          Included file:  lens.h


C  Perform it iterations of block SOR on the lens.
C  Returns error estimate determined from maximum
C  absolute change over this block of iterations.
C  Leaves final result in UA, UB.
C  Update sequence is
C        Ared, Ablk, Xred, Xblk, Bred, Bblk.

      real function SORlens(it)
      integer it
      INCLUDE 'lens.h'
      real, array(0:Ax,0:Rx) :: UA1, UA2, KAp, KAm
      real, array(0:Bx,0:Dx) :: UB1, UB2, KBp, KBm
      integer j
      real qw, onwew, sixthw, mxa, mxb
      qw = 0.25*W          ! For convenience
      sixthw = w/6.0
      onew = 1.0 - W
      UA1 = UA             ! Initialize everywhere
      UA2 = UA
      where ( KA > 0 )     ! For cylindrical coords
       KAp = 0.5/KA
      elsewhere
       KAp = 0.0
      end where
      KAm = 1.0 - KAp      ! ( 1 - 1/2k)
      KAp = 1.0 + KAp      ! ( 1 + 1/2k)
      UB1 = UB
      UB2 = UB
      KBp = 0.5/KB  ! For cylindrical coords
      KBm = 1.0 - KBp      ! ( 1 - 1/2k)
      KBp = 1.0 + KBp      ! ( 1 + 1/2k)
      do, j = 1, it
        where ( Ared )        ! A interior
          UA2 =  KAp*CSHIFT(UA1,2, 1)
     *         + KAm*CSHIFT(UA1,2,-1)
          UA2 = UA2 + CSHIFT(UA1,1,1) + CSHIFT(UA1,1,-1)
          UA1 = qw*UA2 + onew*UA1
        end where
        where ( Ablk )
          UA2 =  KAp*CSHIFT(UA1,2, 1)
     *         + KAm*CSHIFT(UA1,2,-1)
          UA2 = UA2 + CSHIFT(UA1,1,1) + CSHIFT(UA1,1,-1)
          UA1 = qw*UA2 + onew*UA1
        end where
        where ( Xred )        ! Axis .
          UA2 = CSHIFT(UA1,1,1) + CSHIFT(UA1,1,-1)
     *        + 4.0*CSHIFT(UA1,2,1)
          UA1 = sixthw*UA2 + onew*UA1
        end where
        where ( Xblk )
          UA2 = CSHIFT(UA1,1,1) + CSHIFT(UA1,1,-1)
     *        + 4.0*CSHIFT(UA1,2,1)
          UA1 = sixthw*UA2 + onew*UA1
        end where
        UB1(0, 1:M-1) = UA1(1:M-1, N-1)  ! Overlap
        where ( Bred )        ! B interior
          UB2 =  KBp*CSHIFT(UB1,1, 1)
     *         + KBm*CSHIFT(UB1,1,-1)
          UB2 = UB2 + CSHIFT(UB1,2,1) + CSHIFT(UB1,2,-1)
```

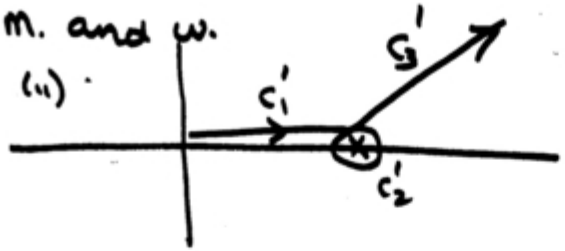
```
      UB1 = qw*UB2 + onew*UB1
    end where
    where ( Bblk )
      UB2 =  KBp*CSHIFT(UB1,1, 1)
*          + KBm*CSHIFT(UB1,1,-1)
      UB2 = UB2 + CSHIFT(UB1,2,1) + CSHIFT(UB1,2,-1)
      UB1 = qw*UB2 + onew*UB1
    end where
    UA1(1:M-1, N) = UB1(1, 1:M-1)     ! Overlap
  end do
  mxa = maxval(abs(UA1 - UA))
  mxb = maxval(abs(UB1 - UB))
  SORlens = max(mxa, mxb)
  print *, "mxa,b, ans ", mxa, mxb, SORlens
  UA = UA1  ! print makes it work
  UB = UB1
  end function SORlens
```

lens.h

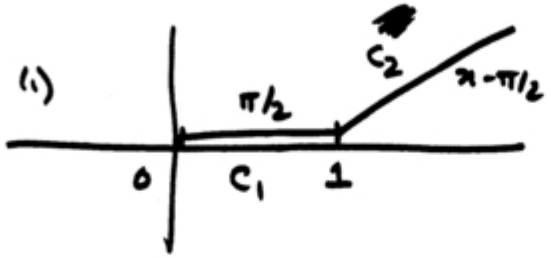
```
C  Parameter and global defn's for lens.f
C  This versions defines small arrays for testing.

      integer, parameter :: Ax = 127, Rx = 63,
     *                      Bx = 127, Dx = 63

      integer M, Mx, N, Nx
      real W
      COMMON /LENSFE/  M, Mx, N, Nx, W

      real UA(0:Ax, 0:Rx), UB(0:Bx, 0:Dx)
      COMMON /LENSPE/ UA, UB
CMF$  LAYOUT UA(,), UB(,)

      integer, array(0:Ax, 0:Rx) :: IA, KA
      integer, array(0:Bx, 0:Dx) :: IB, KB
      COMMON /LENSPE/ IA, KA, IB, KB
CMF$  LAYOUT IA(,), KA(,), IB(,), KB(,)

      logical, array(0:Ax, 0:Rx)::Ared, Ablk, Xred, Xblk
      logical, array(0:Bx, 0:Dx)::Bred, Bblk
      COMMON /LENSPE/ Ared, Ablk, Xred, Xblk, Bred, Bblk
CMF$  LAYOUT Ared(,), Ablk(,), Xred(,), Xblk(,)
CMF$  LAYOUT Bred(,), Bblk(,)

Cnnnn*SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

m. and w.

(a) (ii)



(i)



in (i) $C_1 \Rightarrow \pi/2$

$C_2 \Rightarrow x - \pi/2$ by def of x

Now in (ii), C_1' is same integrand as C_1 $\therefore$ get $\pi/2$

C_2' is zero in limit of vanishing radius ($\int_{\substack{\text{small} \\ \text{circle} \\ |x|=R}} 1/\sqrt{x} \sim 2\pi R/\sqrt{R}$ $\to 0$ as $R \to 0$

C_3' has <u>opposite</u> sign to integrand compared with C_2

$\therefore C_3' \Rightarrow -(x - \pi/2)$

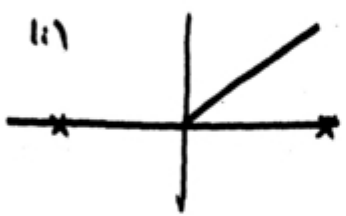
But we get to same value of y.

Hence $\sin(x)$ — (i)

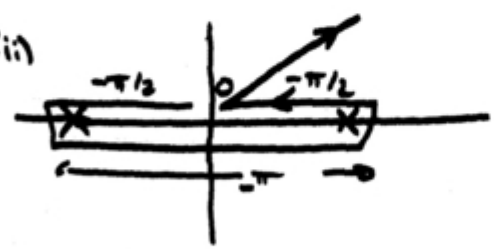
$= \sin\left(\pi/2 + 0 - (x - \pi/2)\right)$ — (ii)

$\equiv \sin(\pi - x)$

(b) (i)



(ii)



In (ii) when we get back to origin we have <u>same</u> sign for integrand as we rounded $\int$ singls twice (picking up − sign each time)

∴ Counting other integrals

$\sin x = \sin(x - 2\pi)$.

Problem Set 2

M + W. problems 7-3, 6, 17, 21, 23, 24

7-21 coeff of y'

$$z(2z^2-1)$$

and 2nd edition

26/Sept/73 Additions.

Grade Marks : ≈ 50% Homework
50% Exams

Grading : Set on Friday
Due 10 days later (
end of Monday class).

75% if handed in by ~~upto week later~~ following Friday.

Anything else – Symbolic credit based on number of questions answered and 50% expected mark. (NOT marked)

Answers in L-D Library 2 weeks after set.

Grader Ted Barnes 422-48, ext 2617

me : G.C. Fox 356-48 ext 2673

Last Half of Course. X

If theres any interest - students can study can study specialized topics. ~~[illegible]~~ instead of coming to class and taking exams.

Either write essay and or take oral/written exam.

A as page 4, 5

plus : more details on functional analysis

$|z| = 1$

$z = 1$ converges $\mathrm{Re}(c-a-b) > 0$
diverges $\mathrm{Re}(c-a-b) \leq 0$.

$|z| = 1,\ z \neq 1$ absolute convergence $\mathrm{Re}(c-a-b) > 0$

converge but not absolutely $-1 < \mathrm{Re}(c-a-b) \leq 0$.

" $-1 = \mathrm{Re}(c-a-b)$
$\mathrm{Re}(a+b-ab) > 0$

diverges $-1 = \mathrm{Re}(c-a-b)$
$\mathrm{Re}(a+b-ab) \leq 0$

diverges $-1 > \mathrm{Re}(c-a-b)$

4 Sept on reserve
Mathews and Walker
Denney & K
Birkhoff & Rota
Morse & Feshbach
Smible

A.M. and W:

C.H.E. 7-29, 7-30, 7-8

Hyper 7-21 → 7.25

Legendre: 7-3

7-6

Bessel: 7-12, 7-17

Asympt 7-14, 3-35, 3-36

Mathieu 7-31, 7-32, 7-33.

WCB 1-42 is a good problem to set
1-43 is also fun.

7-21 → 24

7.21, 7-24, 9.1

↓ error

$z(2z^2-1)y'$

↓

Ph 129: Mathematical Physics

Introduction

This course is officially based on "Mathematical Methods of Physics" by J. Mathews and R.L. Walker. Hereafter M. and W. in the notes, I will deviate quite a lot from this as I prefer to cover fewer topics than in M. and W. - but treat each chosen topic more deeply. M. and W. is ~~an excellent~~ a very readable introduction to a plethora of subjects - it is convenient to discuss the contents of my course by reference to their table of contents.

Chapter 1 Ordinary Differential Equations
2 Infinite Series
3 Evaluation of Integrals
4 Integral Transforms
5 Complex Variables
6 Vectors and Matrices
7 Special Functions
8 Partial Differential Equations
9 Eigenfunctions and Greens Functions
10 Perturbation Theory
11 Integral Equations.
12 Calculus of Variations.
Appendix Introduction to Complex Variables

These first 12 chapters are rather classical and probably familiar to greater or lesser degree to the student. We will not cover, at all, chapters 2, 3 and the appendix: these are vital techniques and the student must read this material if he is at all unsure.

We will start with chapters 1 and 7 which are essentially methods of solving second order linear differential equations and use of special functions as examples. Then we will cover an upgraded version of chapter 6 developing the ideas of function spaces and generalized functions. This can be illustrated by the various sets of special polynomials (Legendre, Hermite ..) found earlier. Given this technical expertise, we will then cover integral equations, Green's functions and partial differential equations (chapters 8 to 11).

Chapters 5 and 12 will not be covered.

They are (relatively) specialized topics which will either be done in quantum physics (dispersion relations) or are probably already known. (calculus of variations / conformal maps). The interested student should read this material.

The second half of the course covers a few specialized topics. M. and W. offer:

Chapter 13 : Numerical Methods.
14 : Probability and Statistics.
15 : Tensor Analysis.
16 : Group theory.

We will cover chapters 13, 14, 16 but leave 15 to a general relativity course.

We will also cover asymptotic expansions (part of chapter 3. in M. and W.) and, if time, other advanced ideas in complex variable theory. [entire functions]

A. Solution of Ordinary Differential Equations and applications to Special Functions. (Practical Aspects)

A1 : Introduction and Omissions

we are discussing the equation:

$$F(x, y, y', \ldots y^{(n)}) = 0 \qquad (A1.1)$$

where $y = y(x)$ & we can define terms ~~[illegible]~~ means :

"ordinary" means y is a function of 1 variable only.

" order " $= n$ is highest derivative present

" degree " is power of highest derivative after equation has been rationalized in $y, y' \ldots y^{(n)}$.

e.g. $y''' + x\sqrt{y} = 0$ has order 3 and degree 2.

" linear " implies equation can be written

in a form linear in $y_1 \ldots y^{(n)}$.

e.g. $(y'''')^2 - x^2 y = 0$ is nonlinear

Putting $L(y) \equiv F(x, y_1 \ldots y^{(n)})$ (A1.2)

A linear DE (differential equation)

has fundamental sum property that:

$$L(y_1 + y_2) = L(y_1) + L(y_2) \qquad (A1.3)$$
$$L(cy_1) = cL(y_1) \quad : c \text{ independent of } x$$

This implies superposition principle:

if y_1 and y_2 satisfy (A1.1), then so does

$c_1 y_1 + c_2 y_2$.

This is sufficient terminology to define

our omissions. The latter are filled in M. and W. and.

E.L. Ince: "Integration of Ordinary Differential Equations". (hereafter Ince).

G. Birkhoff and G-C Rota: "Ordinary Differential Equations" (hereafter B. and R.).

P. Dennery and A. Krzywicki, "Mathematics for Physicists". (hereafter D. and K.)

P.M. Morse and H. Feshbach, "Methods of Theoretical Physics" (Hereafter M. and F.)

We will treat in A2, power series solutions of second order linear DEs (M. and W chapter 1-2; Ince chapter 6; B. and R. chapters 3 and 9).

A4 will be the WKB method (M. and M. chapter 1-4).

We will skip a) sundry elementary techniques. e.g. ~~variation of parameters, integrating factors. (M. and W chapters 1-1, 1-3, Ince~~

e.g. variation of parameters, integrating factors, homogeneous equations etc.

(M. and W chapters 1-1, 1-3; Ince chapters 1,3 and 5; B. and R. chapters 1,2).

b) Existence and uniqueness theorems. (B. and R. chapter 5) – although these will be discussed in sections B and C.

c) Solution of linear differential equations of any order with constant

constants (the operator D), or more generally Laplace transform techniques. (M. and W. chapter 4-5; Ince chapter 5; B. and R. chapter 4).

We will treat some simple nonlinear equations in A3. (B. and R. chapter 6 §7, Ince chapter 2 and §51).
Sections 7b ~~and 8~~ will cover chapters 9 → 11 of B. and R. and much later we will do chapters 7 and 8 in the numerical analysis section.

A2 Special Functions and Solution of Second Order Linear DEs in Series

A2/1 Formal Theory : Solution in Series

See: M. and W. chapter 1-2.
Ince chapter 6
B. and R. chapters 3 and 9.
D. and K. chapter 4, section 14.

If we write our DE as:

$$y'' = p(x)\, y' + q(x)\, y \qquad (A2/1.1)$$

Suppose p and q are analytic at $x = x_0$ and hence differentiable any number of times. Then x_0 is called an <u>ordinary</u> point of the equation. Given values of y and y' at x_0, then (A2/1.1) gives $y''(x_0)$: differentiating our below equation

$$y''' = py'' + p'y' + qy' + q'y$$

shows that now y, y', y'' give $y'''(x_0)$. This procedure may obviously be extended to find unambiguously all the derivatives of y at x_0. y can now be found in a neighbourhood of [redacted] as a Taylor Series

$$y(x) = y(x_0) + y'(x-x_0) + y''(x-x_0)^2/2 + \cdots$$

but we have not shown it converges. In fact one can do this and, mirabile dictu, y is analytic at x_0. (See B. and R., D. and K.)

If p and/or q are not analytic at x_0 but have an (isolated) singularity, then x_0 is a <u>singular</u> point of the equation. One case is if at x_0,

$$p = p_1(x)/(x-x_0), \quad q = q_1(x)/(x-x_0)^2$$

where p_1 and q_1 are regular at $x = x_0$. In this case x_0 is a <u>regular singular</u> point and an equation with only regular singular points is called <u>Fuchsian</u>. Such equations often crop up and always have <u>one</u> solution of the form

$$y(x) = \sum_{n=0}^{\infty} c_n (x-x_0)^{n+s}$$

and the second solution is either also of this form or closely related to it. (need $\partial/\partial s$ in degenerate cases)

We follow M. and W. at this stage, and use the example of Bessel's equation.

$$x^2 y'' + x y' + (x^2 - m^2) y = 0 \qquad (A2/1.2)$$

Clearly $x = 0$ is a regular singular point, and we try $y = x^s \sum_{n=0}^{\infty} c_n x^n$

$$\sum_{n=0} \left\{ c_n (n+s)(n+s-1) x^{n+s} + c_n (n+s) x^{n+s} + c_n x^{n+s+2} - m^2 c_n x^{n+s} \right\} = 0$$

or equating coefficients of like powers of x to zero we get

$$c_{n-2} + c_n \left[(n+s)^2 - m^2 \right] = 0 \quad , n = 0, 1 \ldots$$

with $c_{-2} = c_{-1} = 0$.

Now put $n = 0$ when we get <u>indicial</u> equation:

$$c_0 \left[s^2 - m^2 \right] = 0.$$

By construction $c_0 \neq 0$ (otherwise redefine s) and so $\underline{s = \pm m}$,

$n = 1$ gives $c_1 = 0$ (except when $m = -\frac{1}{2}$ when as shown in M. and W. we can always put $c_1 = 0$) as $s = \pm \frac{1}{2}$ give

two Solutions.) general n gives

$n=2k+1$ odd: $C_{2k+1} = 0$

n even: $C_{n+2}/C_n = \dfrac{-1}{(n+2)(2s+n+2)}$

and Solution is:

$$y(x) = c_0 x^s \left\{ 1 - \frac{x^2}{4(s+1)} + \frac{x^4}{4.8(s+1)(s+2)} - \cdots \right\}$$

Taking $s = \pm m$ gives two solutions. In degenerate case m is an integer, $s = -|m|$ is infinite. Then $s = |m|$ is one Solution and

$$\partial/\partial s \left[(s+|m|)\, y^{PTO}(x,s) \right]_{s=-|m|} \qquad m \neq 0$$

or $\partial/\partial s \; y(x,s)\Big|_{s=0}$ for $m=0$

gives a second solution. This is almost obvious for in each case

PTO $\left(x^2 d^2/dx^2 + x\, d/dx + (x^2 - m^2) \right) (s+|m|)\, y(x,s)$: $m \neq 0$

$y(x,s)$: $m=0$

is proportional to $(s+|m|)^2$

$y(x,s)$ defined by

$$y(x,s) = x^s \left\{ 1 - \frac{x^2}{(s+2+m)(s+2-m)} + \cdots \right\}$$

as $$Ly = \sum_n x^{n+s} \left\{ c_{n-2} + c_n \left([n+s]^2 - m^2 \right) \right\}$$

and $c_{n-2} + c_n \left([n+s]^2 - m^2 \right) = 0 \quad : n > 0$

so $$Ly = c_0 x^s \left(s^2 - m^2 \right)$$

and so, indeed, derivative wrt s also satisfies equation. These techniques are valid for the general DE at a regular singularity.

Rather than do further examples we will now jump to chapter 7 in Mathews and Walker and study special functions. These can be defined in many ways: D. and K. use "generalized Rodrigues formula" (page 205) but this seems slightly artificial. Simplest is the classical approach using differential equations and power series solutions thereto. We will not cover Legendre or Bessel functions (M. and W. chapters 7-1, 7-2). Rather we start with hypergeometric equation, and just set problems on Legendre/Bessel functions. Our excuse for this strategy is

(a) L/B functions are probably familiar.

(b) L/B " " special cases of the hypergeometric equation and its close friend the confluent hypergeometric equation.

In any case, the student should read ~~the~~ ~~also~~ Sections 7-1 and 7-2 of M. and W.

A2/2 The Hypergeometric Equation

(a) Historical Details

People in physics are always sad that a new particle is not discovered every day and if a week goes by without a fundamental ~~b~~ ough consider the subject dead. It is thus instructive to note how long it took many great mathematicians to develop such a "trivial" concept as the hypergeometric equation. There is an interesting account of this in L. J. Slater "Generalized Hypergeometric Series" - Cambridge (1966). As we shall see, this could be subtitled "Eulogy to an English Clergyman". (In her case - W.N. Bailey). We are discussing the standard generalized hypergeometric series

$$ {}_2F_1[a,b;c;z] = 1 + \frac{abz}{c} + \frac{a(a+1)b(b+1)}{c(c+1)}\frac{z^2}{2!} + \dots \quad (A2/2.1) $$

which is also called the Gauss Series.

Its genesis may be traced to 1655 when brilliant Oxford professor John Wallis (1616–1703) in his Arithmetica Infinitorum, uses hyper (Greek for above or beyond) geometric to denote any Series which is beyond the ordinary geometric Series $1 + x + x^2 + \ldots$ (A2/2.2)

(we could go further back and find a common origin for hyperdermic but we leave that for those who leave physics for medical school).

Wallis Studied: (using modern notation)

$$ {}_2F_0\,[a,1;1] = 1 + a + a(a+1) + \ldots \qquad (A2/2.3) $$

which is infinite unless a is a negative integer.

Progress was fitful during the next 150 years. – Euler (1707–1783) proved in 1748

$$ {}_2F_1\,[-n, b; c : z] = (1-z)^{c+n-b}\,{}_2F_1\,[c+n, c-b; c : z] \qquad (A2/2.4) $$

while in 1770, that romantic Frenchman Vandermonde states his extension of the

binomial theorem.

i.e. $${}_2F_1[-n, b; c: z] = \frac{(c-b)(c-b+1)\ldots(c-b+n-1)}{c(c+1)\ldots(c+n-1)} \quad (A2/2.5)$$

where Euler and Vandermonde cowardly take $a=-n$ to make series finite. [n = integer]. Remember the binomial theorem is

$${}_2F_1[-n, b; b: 1] = [1-(z=1)]^n = 0 = 1 - {}_nC_1 + \ldots \quad (A2/2.6)$$

The next 40 years saw slow progress as the Goettingen School under C.F. Hindenberg (1741-1808) wasted much time (fortunately they were not AEC supported and so Vietnam buildup was unaffected) on various complicated extensions of the multinomial and binomial theorems.

All this was changed on January 20 1812 when C.F. Gauss (1777-1855) delivered his famous thesis: "Disquisitiones generales circa Serien Infinitam" before the Royal Society in Goettingen. Gauss introduced the modern name $F(a, b; c: z)$ for (A2/2.1) and proved

$$ {}_2F_1(a,b;c:z=1) = \frac{\Gamma(c)\,\Gamma(c-a-b)}{\Gamma(c-a)\,\Gamma(c-b)} \qquad (A2/2.7) $$

of which Vandermonde's theorem is a special case. Soon on February 12th of the same year Gauss discussed the convergence of the series.

Progress lapsed until 1836 when E.E. Kummer (1810-1893) first used "hypergeometric" for (A2/2.1) only and showed it satisfied the DE:

$$ z(1-z)\,d^2y/dz^2 + [c-(1+a+b)z]\,dy/dz - aby = 0 \qquad (A2/2.8) $$

which - mirabile overkill - has 24 solutions in terms of Gauss functions of sundry arguments.

In 1857 Riemann (1826-66) extended this theory to any ODE with three regular singularities and introduced his P function which isn't very useful.

In 1879, J. Thomae tidied up loose ends.

meanwhile S. Pincherle (1853–1936) used gamma functions in contour integrals and R. Mellin and E.W. Barnes developed this ~~to~~ ~~prove~~ so that in 1910 the latter proved:

$$\frac{1}{2\pi i}\int_{-i\infty}^{+i\infty} \Gamma(a+s)\,\Gamma(b+s)\,\Gamma(c-s)\,\Gamma(d-s)\,ds$$

$$= \frac{\Gamma(a+c)\,\Gamma(a+d)\,\Gamma(b+c)\,\Gamma(b+d)}{\Gamma(a+b+c+d)} \qquad (A2/2.9)$$

which is an integral analogue of (A2/2.7)

The generalized hypergeometric function

$${}_AF_B\left[a_1,\ldots a_A;\, b_1,\ldots b_B;\, z\right]$$

$$= 1 + \frac{a_1\cdots a_A}{b_1\cdots b_B}\,\frac{z}{1!} + \frac{a_1(a_1+1)\cdots}{b_1(b_1+1)\cdots}\,\frac{z^2}{2!} + \cdots \qquad (A2/2.10)$$

was introduced by Clausen in 1828 for the special case $A=3, B=2$. The theory of (A2/2.10) is most pretty for $z=1$ and this case was studied and preached upon by many an English clergyman: it was perfected by W.N. Bailey in 1920–50.

It is astounding to find L. J. Rogers saying at this time "Nothing remains to be done in the field of hypergeometric series" – such naivety when the whole field of ${}_AF_B[\ldots : z \neq 1]$ was still untouched.

Is such mathematics useful in Physics? In 1958, Regge (tall, red and famous in high energy physics) discussed symmetries of complex clebsch-gordon coefficients for SU_2 in a neat way. Unbeknown to him, SU_2 Clebsch Gordon coefficients are just ${}_3F_2[\ldots : z=1]$ functions and ~~the also~~ he had ~~just~~ reproduced (part of) the results of Thomae (1879) and Whipple (1923). More seriously, it is interesting to note that the group theoretical definition leads trivially to well known orthogonality relations for SU_2 c-g coefficients that are <u>bilinear</u> in them. I have scoured the literature but these have <u>not</u> been found in a

hundred years of study of ${}_3F_2[\ldots : z=1]$ from the series definition (A2/2.10). On the other hand some of symmetry relations I mentioned above e.g. $C(j_1, j_2, j : m_1, m_2) = \pm C(j_2, j_1, j : m_2, m_1)$, are trivial group theoretically but others, e.g. $C(j_1, -j-1-t, -j-1 : m_1, m_2)$ is related to $C(j_1, j+t, j : m_1, m_2)$, are unclear from group theory. The mathematical theory, using (A2/2.10), unifies all the symmetries. Similar remarks may be made about SU_2 6-j coefficients which are called Saal Schutzian ${}_4F_3$'s in the mathematical literature.

We will briefly treat (A2/2.10) in the following but to conclude the history, we note that Heine in 1893 put

$$a_q = \frac{1-q^a}{1-q} \qquad \text{(A2/2.11)}$$

which is a "basic number" such that as

$q \to 1$, $a_q \to a$. So we define a "basic hypergeometric series" by replacing $a, b \ldots$ by $a_q, b_q \ldots$ in (A2/2.1,10). We will not consider this further and can only ~~note~~ leave with Slater's ~~humor~~

"Nevertheless there are many quiet corners of the subject, like this one, which have given much pleasure and intellectual delight to many mathematicians during the past 2 centuries".

(b) Back to Grindstone

One uses special functions quite often - they are lamps to guide one while travelling in uncharted mathematical territory. One should try to gain enough familiarity with them to be able to look up detailed properties in

M. Abramowitz and I. Stegun "Handbook of Mathematical Functions".

and the "Bateman Manuscript" (A. Erdelyi et al.)

M. and W. chapter 7-3 is quite good for the ordinary hypergeometric equation (A2/2.1) which we repeat

$$ {}_2F_1[a,b;c:z] = F(a,b;c:z) $$
$$ = 1 + \frac{ab}{c} z + \frac{a(a+1)\,b(b+1)}{c(c+1)} \cdot z^2/_{2!} + \cdots $$
$$ = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n}{(c)_n \; n!} z^n \qquad \Big\} \text{(A2/2.22)} $$

where $(a)_n = a(a+1)\ldots(a+n-1) = \Gamma(a+n)/\Gamma(a)$

(c) Convergence

Write $F = F(a,b;c:z) = \sum_{n=0}^{\infty} u_n$

Then $u_{n+1}/u_n = \frac{(a+n)(b+n)}{(c+n)(1+n)} z \to z$ as $n \to \infty$.

It follows from D'Alembert's test that the series converges for $|z|<1$ and diverges for $|z|>1$.

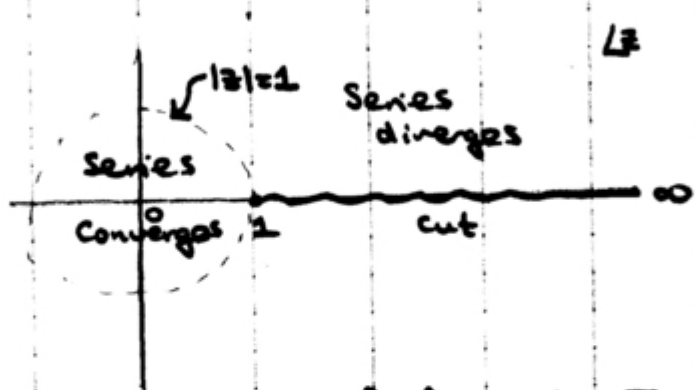
For $|z|=1$ one has to be more careful and expand $u_{n+1}/u_n = 1 + c_1/n + c_2/n^2 + \ldots$

~~Dand~~ Doing this one finds that

$F(a, b; c : z=1)$ converges if

$\mathrm{Re}(c-a-b) > 0$ and if this is true its value is given by Gauss's theorem (A2/2.7).

We will later find that in complex z-plane



and find a representation of F valid outside circle of convergence. (generalization of Euler's transformation (A2/2.4) - this time relating ${}_2F_1$'s of different argument).

(d) Differential Equation

Let $\Delta = z \frac{d}{dz}$

$$(\Delta + a) F = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n z^n}{n!\,(c)_n} \cdot (n+a)$$

$$(\Delta+a)(\Delta+b) F = \quad \text{ditto} \quad (n+a)(n+b)$$

or multiplying by z and replacing $n+1$ by n

$$z(\Delta+a)(\Delta+b)F = \text{ditto } n(n+c-1)$$

but directly

$$\Delta(\Delta+c-1)F = \text{ditto } n(n+c-1)$$

So

$$\Delta(\Delta+c-1)F = z(\Delta+a)(\Delta+b)F$$

or

$$z(1-z)\,d^2y/dz^2 + [c-(1+a+b)z]\,dy/dz - aby = 0 \qquad \text{(A2/2.13)}$$

is DE satisfied by F.

Note this is a second order linear DE with regular singularities at $z=0$ and 1. Now we must define what we mean by nature of point at ∞. This is defined by " " is nature of point $w=0$ after transformation $w = 1/z$. Doing this to (A2/2.13) gives

$$\frac{1}{w}\left(1-1/w\right)\left[2w^3\,dy/dw + w^4\,d^2y/dw^2\right] \qquad \text{(A2/2.14)}$$

$$-\left[c-(1+a+b)/w\right]w^2\,dy/dw - aby = 0$$

or

$$w^2(w-1)\,d^2y/dw^2 + [(2-c)w-(1-a-b)]\,w\,dy/dw - aby = 0$$

which shows that $w=0$ - hence $z=\infty$ - is

a regular singularity.

So it follows that (A2/2.13) has three regular singular points and the rest of the z-plane are ordinary points. The converse is true: any equation with only three regular singular points is essentially the hypergeometric equation. Let us turn to this - so as to both understand solution in series - and also to gain a feeling for Kummer's 24 solutions.

(e) Riemann's Tour de Force (P-function)

Suppose the DE

$$y'' + P(z)\,y' + Q(z)\,y = 0 \qquad \text{(A2/2.15)}$$

has 3 regular singular points $z = z_1, z_2, z_3$.

It follows that:

$$P(z) = \text{Polynomial}(z) / [(z-z_1)(z-z_2)(z-z_3)]$$

$$Q(z) = \text{Polynomial}(z) / [\text{ditto}]^2 \qquad \text{(A2/2.16)}$$

But - remember (A2/2.14) manipulations - we must still impose condition that $z = \infty$ is an ordinary

point. Again we do this by the transformation

$$\omega = 1/z \quad : \quad d\omega = -\omega^2 dz$$

$$y' = -\omega^2 dy/d\omega \quad , \quad y'' = 2\omega^3 dy/d\omega + \omega^4 d^2y/d\omega^2$$

Put $P(z) = \tilde{P}(\omega)$, $Q(z) = \tilde{Q}(\omega)$

Then (A2/2.15) reads

$$d^2y/d\omega^2 + \frac{1}{\omega^2}\left[2\omega - \tilde{P}(\omega)\right] dy/d\omega + \tilde{Q}(\omega)/\omega^4 \, y = 0$$

and the condition that $\omega = 0$ is an ordinary point implies that

$$\tilde{P}(\omega) = 2\omega + O(\omega^2)$$
$$\tilde{Q}(\omega) = O(\omega^4) \qquad \text{(A2/2.17)}$$

Combining (A2/2.16 and 17) gives after manipulation.

$$P(z) = \frac{A}{z-z_1} + \frac{B}{z-z_2} + \frac{C}{z-z_3}$$

with $A+B+C=2$ (A2/2.18)

and

$$Q(z) = \frac{1}{(z-z_1)(z-z_2)(z-z_3)}\left\{\frac{D}{z-z_1} + \frac{E}{z-z_2} + \frac{F}{z-z_3}\right\}$$

Given $P(z)$, $Q(z)$ defined above, we now try to solve (A2/2.15) in series about any one of the regular singularities – say $z = z_1$.

So try:

$$y(z) = (z-z_1)^s \cdot \sum_{m=0}^{\infty} c_m (z-z_1)^m \qquad (A2/2.19)$$

and put to zero the coefficients of $(z-z_1)^{s+m-2}$.

$m=0$ gives the indicial equation

$$s(s-1) + As + \frac{D}{(z_1-z_2)(z_1-z_3)} = 0. \qquad (A2/2.20)$$

Let the roots of (A2/2.20) be $s = \alpha$ and α', so that solutions behave like $y(z) \sim (z-z_1)^{\alpha}$ and $(z-z_1)^{\alpha'}$ near $z = z_1$. α and α' satisfy

$$\alpha + \alpha' = 1 - A$$
$$\alpha\alpha' = \frac{D}{(z_1-z_2)(z_1-z_3)} \qquad (A2/2.21)$$

Now examine solutions near $z = z_2$ and z_3 and define roots β, β' and γ, γ' respectively. Using (A2/2.21) and similar equations for $\beta, \beta', \gamma, \gamma'$ we eliminate $A \to F$ from (A2/2.15,18) to give Pappenty's equation

$$d^2y/dz^2 + \left[\frac{1-\alpha-\alpha'}{z-z_1} + \frac{1-\beta-\beta'}{z-z_2} + \frac{1-\gamma-\gamma'}{z-z_3}\right] dy/dz$$
$$= \theta\varphi\psi y \left[\frac{\alpha\alpha'}{\theta(z-z_1)} + \frac{\beta\beta'}{\varphi(z-z_2)} + \frac{\gamma\gamma'}{\psi(z-z_3)}\right] \frac{1}{(z-z_1)(z-z_2)(z-z_3)} \qquad (A2/2.22)$$

with $\theta = z_2 - z_3$, $\varphi = z_3 - z_1$, $\psi = z_1 - z_2$ (A2/2.23)

The six parameters $\alpha \to \gamma'$ are not independent. The condition (A2/2.18) that $z = \infty$ is ordinary point gives

$$\alpha + \alpha' + \beta + \beta' + \gamma + \gamma' = 1 \qquad \text{(A2/2.24)}$$

The solution of (A2/2.22) is written in terms of Riemann's P-Symbol as:

$$y(z) = P\left\{ \begin{matrix} z_1 & z_2 & z_3 & \\ \alpha & \beta & \gamma & : z \\ \alpha' & \beta' & \gamma' & \end{matrix} \right\} \qquad \text{(A2/2.25)}$$

It is customary to let the P symbol represent the class of <u>all</u> solutions to the DE (A2/2.22) and not just one particular one.

(f) Relation of P-Symbol and Hypergeometric Series

It is quite easy to show that one solution to (A2/2.22) is

$$y(z) = (z - z_1)^{\alpha} (z - z_2)^{-\alpha - \gamma} (z - z_3)^{\gamma} \qquad \text{(A2/2.26)}$$

$${}_2F_1\left[\alpha + \beta + \gamma, \alpha + \beta' + \gamma; 1 + \alpha - \alpha', : \frac{(z - z_1)(z_3 - z_2)}{(z - z_2)(z_3 - z_1)}\right]$$

where γ' does not appear explicitly in (A2/2.26) : however y does depend implicitly on γ' through the constraint (A2/2.24)

We prove (A2/2.26) as follows : by comparing (A2/2.22 and 13) we find that

$$ {}_2F_1(a, b; c : w) \in P \left\{ \begin{matrix} 0 & \infty & 1 \\ 0 & a & 0 \\ 1-c & b & c-a-b \end{matrix} : w \right\} \quad \text{(A2/2.27)} $$

or

$$ {}_2F_1(\alpha+\beta+\gamma, \alpha+\beta'+\gamma ; 1+\alpha-\alpha' : w) $$

$$ \in P \left\{ \begin{matrix} 0 & \infty & 1 \\ 0 & \alpha+\beta+\gamma & 0 \\ \alpha'-\alpha & \alpha+\beta'+\gamma & \gamma'-\gamma \end{matrix} : w \right\} $$

(Note we use $\in$ (belongs to) rather than $=$ as P is class of all solutions)

Now put $$ w = \frac{(z-z_1)(z_3-z_2)}{(z-z_2)(z_3-z_1)} \quad \text{(A2/2.28)} $$

So that $w = \begin{matrix} 0 \\ 1 \\ \infty \end{matrix}$ becomes $z = \begin{matrix} z_1 \\ z_3 \\ z_2 \end{matrix}$

But (A2/2.28) clearly preserves the singularity structure of functions i.e.

if $f(w)$ satisfies a second order DE with 3 regular singularities, then so does $f(z)$.

So with w given by (A2/2.28) we get

$$ {}_2F_1(\alpha+\beta+\gamma,\ \alpha+\beta'+\gamma\ ;\ 1+\alpha-\alpha' : w) $$

$$ \in P\left\{\begin{matrix} z_1 & z_2 & z_3 & \\ 0 & \alpha+\beta+\gamma & 0 & : z \\ \alpha'-\alpha & \alpha+\beta'+\gamma & \gamma'-\gamma & \end{matrix}\right\} \qquad \text{(A2/2.29)} $$

But it is a general property of the P-symbol that

$$ (z-z_1)^{\delta}\ P\left\{\begin{matrix} z_1 & z_2 & z_3 & \\ \alpha & \beta & \gamma & : z \\ \alpha' & \beta' & \gamma' & \end{matrix}\right\} \qquad \text{(A2/2.30)} $$

$$ = \text{"}P\left\{\begin{matrix} z_1 & z_2 & z_3 & \\ \alpha+\delta & \beta & \gamma & : z \\ \alpha'+\delta & \beta' & \gamma' & \end{matrix}\right\}\text{"} $$

for the multiplication by $(z-z_1)^{\delta}$ does not destroy either nature of singularity (only increasing z_1 exponents by δ) or property of satisfying a second

order linear DE. However it does spoil fact that $z=\infty$ is an ordinary point and we only get a true P-function on the right hand side of (A2/2.30) by applying ~~bras~~ multiplication $(z-z_1)^{\delta_1}(z-z_2)^{\delta_2}(z-z_3)^{\delta_3}$ with $\delta_1+\delta_2+\delta_3=0$.

Taking $\delta_1=\alpha$, $\delta_2=-\alpha-\gamma$, $\delta_3=\gamma$ to multiply (A2/2.29) gives finally (A2/2.26)

Q.E.D.

(g) Kummer's Multitude of Solutions

Given one solution, i.e. (A2/2.26), for the P-symbol in terms of a Gauss function we can get 3 more solutions by changing $\alpha \to \alpha'$ or $\gamma \to \gamma'$ for this leaves DE unchanged.

Further given 4 such solutions for a fixed ordering of z_1, z_2 and z_3 we ~~get~~ can apply $3!=6$ permutations of the three z_i to get 24 solutions of

(A2/2.22) in terms of Gauss functions.

These 24 solutions were found by Kummer in 1836 for the hypergeometric equation (A2/2.13) itself. As above these 24 solutions fall into 6 groups of 4: each group being ${}_2F_1$'s of ~~argument~~ a given argument – the 6 arguments are $z, 1/z, 1-z, 1/(1-z), (z-1)/z, z/(z-1)$.

Consider the first group: then $\alpha = 0$ and $\alpha' = 1-c$ are the solutions of the indicial equation at $z=0$. The basic solution is

$$y_1 = {}_2F_1(a, b; c : z)$$

Swopping $\alpha \leftrightarrow \alpha'$ gives

$$y_2 = z^{1-c}\, {}_2F_1(1+a-c, 1+b-c; 2-c : z)$$

All the other 22 solutions must of course be linear combinations of these. However because they have (in general) different argument they converge in

a different part of the complex z plane and provide an analytic continuation of ${}_2F_1$ outside its circle of convergence $|z|=1$. e.g. the solutions with argument $1-z$ converge for $|1-z|<1$ etc.

Taking $\gamma \to \gamma'$ in y_1 gives third solution

$$y_3 = (1-z)^{c-a-b}\, {}_2F_1(c-a, c-b; c : z)$$

But $y_3 = Ay_1 + By_2$ is linearly dependent on y_1 and y_2 and as of y_1, y_2, y_3, only y_2 is singular at $z=0$, we must have $B=0$ whence putting $z=0$ gives $A=1$.

Thus $y_2 = y_3$

which is Euler's identity (A2/2.4) 1748 a=-n

There are many other such relations between $y_1 \ldots y_{24}$: these plus a discussion of special cases when roots of indicial equation differ by an integer, (cf. end of (A2/1)) will be found in Bateman manuscript and Abramowitz and Stegun.

(h) Special cases of the hypergeometric Series

(i) Standard ancestor is

$$ {}_2F_1\left[a, b; b : z\right] = \sum_{n=0}^{\infty} \frac{(a)_n}{n!} \frac{(b)_n}{(b)_n} z^n $$

$$ = [1-z]^{-a} $$

the binomial Series.

(ii) Again

$$ {}_2F_1\left[1, 1; 2 : -z\right] = \sum_{n=0}^{\infty} \frac{(1)_n}{(1)_n} \frac{(1)_n}{(2)_n} (-z)^n $$

$$ = \sum_{n=0}^{\infty} (-z)^n / n+1 $$

$$ = \log(1+z)/z $$

or

$$ {}_2F_1\left[\tfrac{1}{2}, \tfrac{1}{2}; \tfrac{3}{2} : z^2\right] = \tfrac{1}{z} \sin^{-1} z $$

and so on for (all) the simple functions

(iii) As listed (ad nauseam) in the Bateman manuscript the Incomplete β function
Complete elliptic integral
Legendre
Chebyshev
Gegenbauer
Jacobi
} polynomials

are all hypergeometric Series. For example

(iv) Legendre Functions as Hypergeometric Series

Legendre's equation is:

$$(1-x^2)\, d^2y/dx^2 - 2x\, dy/dx + n(n+1)y = 0 \qquad (A2/2.31)$$

clearly this has three regular singularities at $x = \infty, \pm 1$. From our general discussion of Papperitz's equation this must be related to a hypergeometric series when we map singularities to 0, 1 and ∞.

So put $z = (1-x)/2$

$$\begin{aligned} x = 1 &: z = 0 \\ -1 &: 1 \\ \infty &: \infty \end{aligned}$$

and (A2/2.32) becomes

$$2 \cdot 2z(1-z)\, d^2y/4dz^2 + 2(1-2z)\, dy/2dz + n(n+1)y = 0$$

Compare this with the hypergeometric equation (A2/2.13).

This gives: $c = 1$
$a+b+1 = 2$
$ab = -n(n+1)$

whence a, b are solts of $q^2 - q - n(n+1) = 0$

or $a = -n$, $b = n+1$.

So P_n / Q_n or l.c. thereof is proportional to ${}_2F_1(-n, n+1; 1 : (1-x)/2)$

But P_n is a polynomial and Q_n isn't.

Also $P_n(1) = 1$.

So $P_n(x) = {}_2F_1[-n, n+1; 1 : (1-x)/2]$

~~where~~ for as $a = -n$, n an integer, the (A2/2.32) hypergeometric series terminates and is a polynomial in x. $Q_n(x)$ is a linear combination of the other (24) solts of the DE (A2/2.13): it is rather horrific and may be inspected in, say, the Bateman manuscript. ~~(See 7a)~~ we can "derive" it upto a constant quite easily.
~~[illegible]~~

In fact $Q_n(x)$ is defined to be one of solutions with a nice behaviour at infinity. From the representation

$$ {}_2F_1(a,b;c:w) \in P\left\{\begin{matrix} 0 & \infty & 1 & \\ 0 & a & 0 & :w \\ 1-c & b & c-a-b & \end{matrix}\right\} $$

it follows that the arbitary solution to the Hypergeometric equation behaves like

$$ w^{-a}\sum_{m=0}^{\infty} a_m w^{-m} + w^{-b}\sum_{m=0}^{\infty} b_m w^{-m} \quad \text{as } w \to \infty $$

for a, b are from P representations solutions of indicial equation at ∞. Plugging in legendre parameters, this makes $P_n(z)$ (n complex) behave as a combination of z^n and z^{-n-1} at ∞.

$Q_n(z)$ is defined to be that solution which behaves like z^{-n-1} at ∞. This fixes it upto an arbitary constant.

Using (A2/2.26) we can write it down as by

$$\mathcal{P}_n(x) = c\,y(z) \qquad z = \frac{1-x}{2}$$

$$y(z) = P\left\{\begin{matrix} \infty & 0 & 1 & \\ +n+1 & 0 & 0 & : z \\ -n & 0 & 0 & \end{matrix}\right\}$$

$$= z^{-n-1}\,{}_2F_1\left\{n+1,\, n+1;\, 2n+2 : 1/z\right\}$$

c is chosen conventionally to give:

$$\mathcal{P}_n(x) = \frac{2^n\left[\Gamma(n+1)\right]^2}{\Gamma(2n+2)\,(x-1)^{n+1}}\,{}_2F_1\left[n+1, n+1; 2n+2 : \frac{2}{1-x}\right\}$$

Note that indices of Legendre equation are equal (at 0) at $z=0$ and 1. According to standard differentiating dodge this means solutions to Legendre's equation behaves like $1, \log z$ or $1, \log(1-z)$

at $z = 0$ and 1 respectively.

$P_n(x)$ has a logarithmic singularity at $x = 0, 1$ all n

$P_n(x)$ is regular $x = 1$ for n integer (when P_n a polynomial) but for complex n has a logarithmic singularity at $x = 1$.

So hypergeometric is very convenient for discussing behaviour of Legendre functions at magic points. Are there any other things for which relation between

P_n/Q_n and hypergeometric series useful? Somewhat - for some general properties of hypergeometric series give at once properties of Legendre functions. e.g.

(α) Recurrence relation between three P_n's (Q_n's) corresponds to standard relation between three associated hypergeometric series. (we will do this in next section).

(β) Again general integral representations /asymptotic expansions of hypergeometric series give the same for Legendre functions

(γ) However the orthogonality relation

$$\int_{-1}^{+1} P_n(x)\, P_m(x)\, dx = 2\,\delta_{nm}/_{2n+1}$$

has no general analogue for hypergeometric series.

In fact, I believe that all the special properties of P_n for integer n are not easily seen from hypergeometric point of view. Better to use group-theory approach ($P_n(\cos\theta)$ are basis functions of irreducible representations of $O(3)$) given in quantum theory or later in this course. Another neat approach (D. and K. page 203 ff.) uses the generalized Rodriguez formula to discuss a whole class of sets of orthogonal polynomials. The grand-daddy Rodriguez formula is

$$P_n(x) = \frac{1}{2^n n!}\left(\frac{d}{dx}\right)^n [x^2-1]^n$$

A similar specialized integer n approach to Legendre functions comes from the generating function $F(h, x)$

$$F(h,z) = \frac{1}{\sqrt{1-2hz+h^2}} = \sum_{n=0}^{\infty} h^n P_n(z)$$

These are discussed in M. and W. chapter 7-1 which both bored and interested students should read. They also discuss the associated Legendre functions which (as special cases of Jacobi polynomials) are also hypergeometric series. (See Abramowitz and Stegun). Main thing is to be familiar enough with all the monsters in our special function zoo that you can look up detailed properties in A. and S. or Bateman. We can't go through all this in class....

(i) Relations between Associated Hypergeometric Series

Functions of the form ${}_2F_1[a+l, b+m; c+n : z]$ for fixed a, b, c, z and variable integers ($>$ or ≤ 0) l, m, n are said to be <u>associated</u>. It can be shown that <u>any</u> three associated functions are ~~related~~ linearly related with coefficients that are polynomial in a, b, c, z. As we said the recurrence relⁿ

$$(n+1)\, P_{n+1}(z) = (2n+1)\, z\, P_n(z) - n P_{n-1}(z) \qquad (A2/2.32)$$

for Legendre polynomials, is a special case of this. The prove of this result is difficult in general. The simplest example is gotten from

$$x = F(a+1, b; c : z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n z^n}{(c)_n \, n!} \cdot \frac{(a+n)}{a}$$

$$y = F(a, b+1; c : z) = \sum_{n=0}^{\infty} \frac{(a)_n (b)_n z^n}{(c)_n \, n!} \cdot \frac{(b+n)}{b}$$

whence

$$ax - by = (a-b) \, F(a, b; c : z)$$

(A2/2.33)

~~The general case is treated in Bateman~~

~~where~~ Further examples are given in the Bateman manuscript. (This concentrates on rel^{ns} between contiguous functions – two hypergeometric functions are contiguous if differ in but one parameter and difference is ± 1 e.g. x and y are contiguous to $F(a, b; c : z)$).

(j) Integral Representations

This is covered well in M. and W. pages 191 to 194. Consider

$$f(x) = \int t^{b-1}(1-t)^{c-b-1}(1-tx)^{-a}\,dt \qquad \text{(A2/2.34)}$$

Then direct evaluation gives

$$x(1-x)f'' + [c-(a+b+1)x]f' - abf =$$

$$\int \Big[a(a+1)x(1-x)t^2 + [c-(a+b+1)x]at - ab \Big] t^{b-1}(1-t)^{c-b-1}(1-tx)^{-a}\,dt$$

$$= -a\int dt\,\frac{d}{dt}\left[t^{b}(1-t)^{c-b}(1-tx)^{-a-1}\right]$$

$$= -a\int dt\,\frac{d}{dt}F(t,x) \quad \text{defining } F(t,x)$$

to be the quantity in [ ].

So to ensure that f satisfies the hypergeometric DE (A2/2.13) we must have

$$\Big[F(t,x)\Big]_{\text{begin contour int}}^{\text{end contour int}} = 0 \qquad \text{(A2/2.35)}$$

There are two (usual) ways of doing this

(i) The simplest is to take in (A2/2.34) the integral as $\int_0^1 dt$

Then if $\mathrm{Re}\, c > \mathrm{Re}\, b > 0$ (A2/2.36)

$$F(0,x) = F(1,x) = 0 \quad \text{and}$$

truely $f(x)$ satisfies (A2/2.13).

Also $f(x)$ is regular at $x=0$ and so f is proportional to ${}_2F_1(a,b;c:x)$.

Putting $x=0$ in (A2/2.34) gives

$$f(0) = \int_0^1 t^{b-1}(1-t)^{c-b-1}\,dt$$

$$= \beta\text{-function integral} = \frac{\Gamma(b)\,\Gamma(c-b)}{\Gamma(c)}$$

Hence

$$f(x)\,\frac{\Gamma(c)}{\Gamma(b)\,\Gamma(c-b)} = {}_2F_1(a,b;c:x) \qquad \text{(A2/2.37)}$$

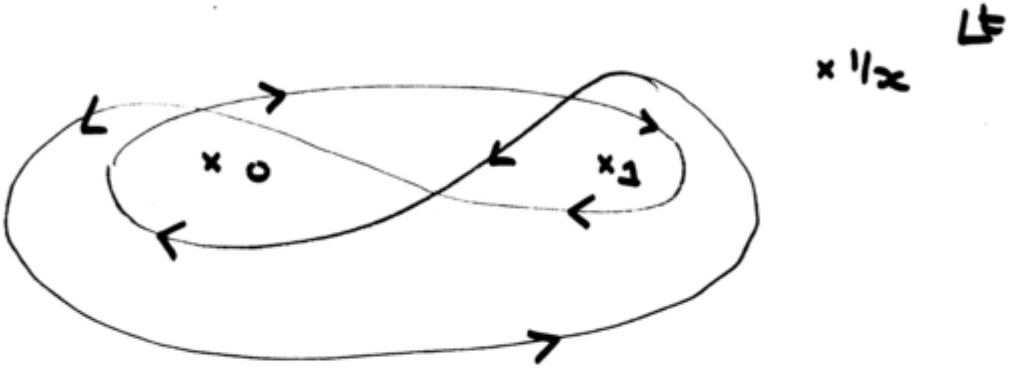
which provides an integral representation for ${}_2F_1$ when (A2/2.36) satisfied.

(ii) A second way of satisfying (A2/2.35) is take a closed contour in t for the integral in (A2/2.34). This will have no restriction (A2/2.35) on b or c but one must ensure that $F(t,x)$ comes back to same value on circling contour ~~function~~ once. Now each time you circle a singularity $(t-t_0)^\alpha$ you pick up a horible phase $\exp[2\pi i\alpha]$. Thus success dictates that each singularity be circled either no times or twice - once in each direction.

If you circle no singularities the result is trivially zero from Cauchy's theorem. But circling one singularity twice i.e.

also gives zero.

So simplest possibility is Pochhammer's contour which circles the singularities $t=0$ and 1 twice.



This time the phases cancel and indeed the singular integral is nonzero.

Where are the singularities of $f(x)$? This can be determined by an interesting general "pinch" technique as when the floating singularity $t = 1/x$ hits a fixed one $t=0,1$. For other values of x one can deform contour away from singularities $t=0,1,1/x$. So pinch mechanism

gives singularities at $x = 1$ and ∞ and $f(x)$ is again proportional to ${}_2F_1(a, b; c : x)$ as both are regular at $x = 0$.

(k) Generalized Hypergeometric Series

It is worthwhile realizing that these exist but I wont go into details - just define them - and reader can consult Bateman manuscript or Slater when his mathematical voyages meet up with one. Define:

$$ {}_AF_B[a_1 .. a_A ; b_1 .. b_B : z] \qquad \text{(A2/2.38)} $$

$$ = \sum_{n=0}^{\infty} \frac{(a_1)_n (a_2)_n \cdots (a_A)_n}{(b_1)_n \cdots (b_B)_n} z^n/n! $$

Putting $\Delta = z\,d/dz$ we can - just as in Section (d) show that ${}_AF_B$ satisfies the DE

$$ \{ \Delta(\Delta+b_1-1) \cdots (\Delta+b_B-1) - z(\Delta+a_1) \cdots (\Delta+a_A) \} y = 0 \qquad \text{(A2/2.39)} $$

if $A=2$, $B=1$ we get back the Gauss function but if $A=B+1$ we ca

Show from (A2/2.39) that DE has again regular singularities at $z = 0, 1$ and ∞. So the analysis is theoretically similar - the equation is just higher order if $A > 2$.

The reference books have several (relatively) obvious generalizations of integral, differential and recurrence relations we discussed for ancestral Gauss function. Also there are a lot of symmetry and summation formulae - the latter generalize especially for $z=1$ Gauss's formula (A2/2.7) namely

$$ {}_2F_1(a, b; c : z=1) = \frac{\Gamma(c)\,\Gamma(c-a-b)}{\Gamma(c-a)\,\Gamma(c-b)} $$

As I mentioned in the historical survey the symmetry relations could have useful for studying SU_2 Clebsch-

Gordon coefficients for complex angular momentum. Thus

$$C(j_1 j_2 j : m_1 m_2) = c$$

$$\cdot\, {}_3F_2\left[-j_1+m_1, -j_2-m_2, -j_1-j_2+j \,;\, j-j_1-m_2+1,\right.$$

$$\left. j-j_2+m_1+1 \,:\, 1\right] \qquad (A2/2.40)$$

where the constant c is enormous and given as:

$c =$

A2/3 The Confluent Hypergeometric Equation

We continue our voyage through chapter 7 of M. and W. and now come to pages 194 to 197. Consider the hypergeometric DE (A2/2.13) - namely

$$x(1-x)\,y'' + [c-(a+b+1)x]\,y' - aby = 0 \qquad \text{(A2/2.13)}$$

and place in it $x = z/b$ so that if

$y(x)$ is singular at $x = 0, 1, \infty$

$y(z)$ " " " $z = 0, b, \infty$, and satisfies

$$z/b\,(1-z/b)\,b^2\,d^2y/dz^2 + [c - (a+b+1)z/b]\,b\,dy/dz - aby = 0$$

Divide equation by b and let $b \to \infty$.

$$z\,d^2y/dz^2 + [c-z]\,dy/dz - ay = 0 \qquad \text{(A2/3.1)}$$

This is confluent hypergeometric equation with a regular singularity at $z=0$ and an <u>essential</u> singularity at $z=\infty$ which is the unholy alliance of those at $z=b$ and ∞.

Our standard solution of (A2/2.13) was

$$y_1 = 1 + ab\,x/c + \cdots + \frac{(a)_n (b)_n x^n}{(c)_n \; n!} + \cdots$$

Now fix n, put $x = z/b$ and let $b \to \infty$. Then $(b)_n x^n \to z^n$ and so solution is

$$y(z) = 1 + \frac{az}{c} + \cdots + (a)_n z^n / (c)_n n! + \cdots$$

(A2/3.2)

$= {}_1F_1[a; c : z]$ using notation of generalized hypergeometric function. Using above limiting procedure we can derive properties of confluent hypergeometric equation from those of the Gauss function. We refer the reader to M. and W./Bateman manuscript for details of integral representations etc.

Finally we note that confluent generalized hypergeometric functions ${}_AF_A[\ldots]$ may be defined using our

limiting procedure on ${}_{n+1}F_n[\ldots]$.

(b) Examples of Confluent Hypergeometric Functions

(i) $${}_1F_1(a\,;a:z) = 1 + \frac{az}{a} + \cdot \quad + \frac{(a)_n}{(a)_n} z^n/n!$$
$$= \exp(z)$$

which illustrates essential singularity at $z=\infty$.

Note ${}_2F_1[a,b\,;a:x] = [1-x]^{-b}$

Putting $x = z/b$ and $b \to \infty$ gives

$$\lim_{b\to\infty} [1 - z/b]^{-b} = \exp(z)$$

a relic of times gone by.

(ii) Again

$$\operatorname{erf}(z) = 2/\sqrt{\pi} \int_0^z \exp(-t^2)\,dt = \frac{2z}{\sqrt{\pi}}\,{}_1F_1(1/2\,;3/2:-z^2)$$

(iii) Bessel-Functions

Bessel's equation - which we have already looked at in A2/1 - reads

$$z^2 u'' + z u' + (z^2 - m^2) u = 0 \qquad \text{(A2/3.3)}$$

Put $u = z^m e^{-iz} f(x)$: $x = 2iz$

which transformation is both hard to perform and tricky to motivate [could by along lines of argument in A2/2(f) but I haven't followed it through]. One obtains:

$$x \, d^2f/dx^2 + [(2m+1) - x] \, df/dx - \frac{(2m+1)}{2} f = 0 \qquad \text{(A2/3.4)}$$

which is the confluent hypergeometric equation (A2/3.1) with $a = m + \frac{1}{2}$ and $c = 2m+1$. One solution is $\Phi(a, c; x) = {}_1F_1(a; c : x)$ and so we can take as one solution of Bessel's equation

$$J_m(z) = \frac{1}{\Gamma(m+1)} \left(z/2\right)^m e^{-iz} \Phi(m+\tfrac{1}{2}, 2m+1; 2iz) \qquad \text{(A2/3.5)}$$

A second solution can be found in two ways:

(α) clearly (A2/3.3) is symmetric on $m \to -m$ and so $J_{-m}(z)$ [i.e substitute m to $-m$ in (A2/3.5)] is also a solution. [~~[illegible]~~ examining behaviour as $z \to 0$ shows that it is in general independent of $J_m(z)$].

(β) we can derive ~~solution~~ second solution of general confluent HE just as we ~~[illegible]~~ did in Kummer's analysis of the ordinary HE. (page 35)

In fact we need only perform the process of confluence on $y_2 = x^{1-c}\, {}_2F_1[1+a-c, 1+b-c; 2-c : x]$ to show that $x^{1-c}\, \Phi(a-c+1, 2-c, x)$ is a second solution to (A2/3.1). Multiplying the latter by $z^m e^{iz}$ gives a second solution to (A2/3.3). This method gives – upto an irrelevant overal constant – the same $J_{-m}(z)$ we found above.

As usual this process breaks down when m is an integer when J_m and J_{-m} become dependent. i.e.

$$J_m(z) = (-1)^m J_{-m}(z) \qquad (A2/3.6)$$

(It is an easy exercise to prove this from the confluent form (A2/3.5). So we introduce <u>Bessel's function of the Second kind</u> or <u>Neumann's Function</u> by:

$$Y_m(z) = \frac{\left[J_m(z)\cos(m\pi) - J_{-m}(z) \right]}{\sin(m\pi)} \qquad (A2/3.6)$$

which are always independent of $J_m(z)$.

Again one often uses:

$$H_m^{(1)}(z) = J_m(z) + i\,Y_m(z) \qquad (A2/3.7)$$

$$H_m^{(2)}(z) = J_m(z) - i\,Y_m(z)$$

which are <u>Hankel functions</u> or <u>Bessel functions of the third kind</u>.

An important property of Bessel functions is their asymptotic behaviour (which is exponential as leftz an essential singularity). We have:

$$J_m(z) \sim \sqrt{\frac{2}{\pi z}} \cos\left(z - \tfrac{1}{2}m\pi - \tfrac{1}{4}\pi\right) \qquad |\arg z| < \pi$$

$$Y_m(z) \sim \text{"} \quad \sin(\quad \text{"} \quad) \qquad |\arg z| < \pi$$

$$H_m^{(1)}(z) \sim \text{"} \quad \exp i(\quad \text{"} \quad) \qquad -\pi < \arg z < 2\pi \qquad \text{(A2/3.8)}$$

$$H_m^{(2)}(z) \sim \text{"} \quad \exp -i(\quad \text{"} \quad) \qquad -2\pi < \arg z < \pi$$

which provides some motivation for the myriad of Bessel functions. Proof of (A2/3.8) (and its extension to other arg z) requires techniques we will cover when we treat the general topic of asymptotic expansions.

There are also modified Bessel functions $I_m(z)$, $K_m(z)$ and even spherical Bessel functions $j_n(z)$, $y_n(z)$, $h_n^{(1)}(z)$, $h_n^{(2)}(z)$

with honor upon honor modified spherical Bessel and Riccati Bessel functions. Also of importance are Airy functions which are related to $J_{\pm 1/3}(z)$. Please explore this jungle in your favorite text book. (Section 7-2 of M. and W.].

Note that just like ${}_2F_1$ (Section (A2/2)(i)) "associated" ${}_1F_1$'s satisfy

a three term linear relation. This leads to the usual recurrence relation for $J_n(x)$.

(iv) Further we find that Hermite polynomials (Schrödinger's Harmonic oscillator equation) and Laguerre polynomials (Schrödinger equation with coulomb potential, Gauss-Laguerre integration) are also related to confluent hypergeometric functions.

The Bateman manuscript has more examples. They even found a "Toronto function" - perhaps realtors should sell one parameter families of (confluent) hypergeometric functions.

A2/6 Gamma Function

For completeness we mention here the Γ-function: references are:

Chapter 6 : Abramowitz and Stegun
pages 419 to 425 : M. and F.
page 75 to 76 : M. and W.

Definition is:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt \qquad \text{(A2/6.1)}$$

which defines $\Gamma(z)$ as an analytic function of z. Integral only valid for $R(z) > 0$ but relation

$$\Gamma(z+1) = z\,\Gamma(z) \qquad \text{(A2/6.2)}$$

extends definition to whole complex plane. We follow with some important properties: the reader should check references if he is not comfortable with them.

Integer n $\quad n! = \Gamma(n+1)$ (A2/6.3)
with $0! = 1$.

$\Gamma(1/2) = \sqrt{\pi}$ ($\approx$ error integral) (A2/6.4)

$\Gamma(z)$ has singularities at $z = -n$

with residue $(-1)^n/n!$. (A2/6.5)

$$\Gamma(z)\,\Gamma(1-z) = \pi/\sin \pi z \qquad \text{(A2/6.6)}$$

Duplication Formula

$$\Gamma(2z) = (2\pi)^{-1/2}\, 2^{2z-1/2}\, \Gamma(z)\,\Gamma(z+1/2) \qquad \text{(A2/6.7)}$$

Stirling's Formula - we will prove later

$$\Gamma(z) \sim \exp(-z)\; z^{z-1/2}\,(2\pi)^{1/2} \qquad \text{(A2/6.8)}$$

for large z.

Beta - Function Integral:

$$B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}\,dt$$

(A2/6.9)

A2/4 Mathieu's Equation

(a) This is studied not only for its alliterative value but also as an example of a differential equation with periodic coefficients. It may be found in.

M. and W. pages 198 to 203
M. and F. page 556 ff.

The equation is:

$$d^2y/dx^2 + [\alpha + \beta \cos 2x]\, y = 0 \qquad (A2/4.1)$$

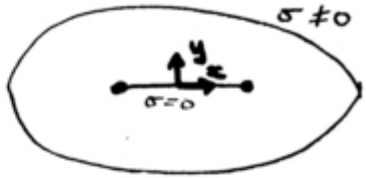
(b) Example of its occurrence

We gladden the hearts of mathematicians of old by considering elliptic co-ordinates. Just in case you haven't used these recently we remind you that these are given by:

$$x = \rho \cosh\sigma \cos\varphi, \quad y = \rho \sinh\sigma \sin\varphi. \qquad (A2/4.2)$$

ρ is fixed and as φ varies for fixed σ

x, y describe the ellipse



$$\frac{x^2}{\rho^2\cosh^2\sigma} + \frac{y^2}{\rho^2\sinh^2\sigma} = 1$$

with foci at $x = \pm\rho$, $y = 0$.

We transform $\nabla^2\Phi$ in our new co-ordinate system as:

$$\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} = \frac{2}{\rho^2[\cosh 2\sigma - \cos 2\varphi]}\left\{\frac{\partial^2\Phi}{\partial\sigma^2} + \frac{\partial^2\Phi}{\partial\varphi^2}\right\} \qquad \text{(A2/4.3)}$$

Now solve

$$\nabla^2\Phi + k^2\Phi = 0 \qquad \text{(A2/4.4)}$$

by trying $\Phi(\sigma,\varphi) = \Phi_1(\sigma)\Phi_2(\varphi)$: Standard seperation of variables. This gives:

$$\frac{1}{\Phi_1} d^2\Phi_1/d\sigma^2 + \frac{1}{\Phi_2} d^2\Phi_2/d\varphi^2 + \frac{\rho^2 k^2}{2}[\cosh 2\sigma - \cos 2\varphi] = 0. \qquad \text{(A2/4.5)}$$

So from the standard argument, the seperate functions of σ and φ in (A2/4.5) must be a constant : say $\pm a$.

So: $$a = \frac{d^2\Phi_1}{d\sigma^2} \cdot \frac{1}{\Phi_1} + \frac{\rho^2 k^2}{2} \cosh\sigma \qquad \text{(A2/4.6a)}$$

<u>or</u> if $\alpha = a$, $\beta = -\rho^2 k^2/2$, $\sigma = ix$

we get back Mathieu's equation

<u>and</u> $$-a = \frac{1}{\Phi_2} \frac{d^2\Phi_2}{d\varphi^2} - \frac{\rho^2 k^2}{2} \cos 2\varphi \qquad \text{(A2/4.6b)}$$

which is directly Mathieu's equation with again $\alpha = a$, $\beta = -\rho^2 k^2/2$. $(x = \varphi)$.

Although artificial, this example is nice because in the "angular equation" (A2/4.6b), the physics demands that we only accept solutions satisfying

$$y(\varphi) = y(\varphi + 2\pi) \qquad \text{(A2/4.7)}$$

However for the "radial equation" - (A2/4.6a) there is no such condition - rather we are interested in the behaviour as $\sigma \to \infty$ where usually we would require $y \to 0$.

These points are clear in the special case β (i.e. $\rho^2 k^2$) $=0$ when we get $\frac{d^2y}{d\varphi^2} + \alpha y = 0$ with exponential solutions for $\alpha < 0$ and periodic solutions $\alpha > 0$. The latter only have periods $2\pi/m$, for special [$\alpha = m^2$] values of α.

Now return to the general case (A2/4.1)

(c) Floquet's Theorem

Let $y_1(x)$ and $y_2(x)$ be any two independent solutions of Mathieu's equation (A2/4.1). Then the periodicity of the coefficients ensures that $y_1(x+2\pi)$ and $y_2(x+2\pi)$ are also solutions. It follows that the latter must be linear combinations of our original choice y_1 and y_2. Thus

$$\begin{aligned} y_1(x+2\pi) &= A_{11}\, y_1(x) + A_{21}\, y_2(x) \\ y_2(x+2\pi) &= A_{12}\, y_1(x) + A_{22}\, y_2(x) \end{aligned} \qquad \text{(A2/4.8)}$$

Floquet's theorem states that we can find a solution satisfying

$$y(x+2\pi) = k\, y(x) \qquad (A2/4.9)$$

we by $y(x) = C_1 y_1(x) \cancel{+ y_2(x)} + C_2 y_2(x)$

Then by explicit calculation

$$y(x+2\pi) = C_1 y_1(x+2\pi) + C_2 y_2(x+2\pi)$$

$$= (C_1 A_{11} + C_2 A_{12}) y_1(x) + (C_1 A_{12} + C_2 A_{22}) y_2(x)$$

and this equals $k\, y(x)$, if last line is

$$= k C_1 y_1(x) + k C_2 y_2(x)$$

or

$$C_1 A_{11} + C_2 A_{12} = k C_1$$
$$C_1 A_{21} + C_2 A_{22} = k C_2 \qquad (A2/4.10)$$

or introducing matrix notation:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$$

(A2/4.10) becomes

$$AC = kC$$

i.e. k/C is eigenvalue/eigenvector of A: but A always has one such eigenvalue. So

This proves k, C_1, C_2 exist and hence completes Floquet's theorem.

<u>Corollary</u> : we can write $y(x)$ of Floquet's theorem in the form:

$$y(x) = \exp[iSx]\,\varphi(x) \qquad \text{(A2/4.11)}$$

with $\quad \varphi(x) = \varphi(x+2\pi)$

<u>Proof</u>: Put $k = \exp[2\pi i S]$ in (A2/4.9) and define $\varphi(x) = \exp[-iSx]\, y(x)$

Then $\varphi(x+2\pi) = e^{-2\pi i S}\, y(x+2\pi)\, e^{-iSx}$

$$= \frac{1}{k}\, k\, y(x)\, e^{-iSx} = \varphi(x)$$

Q.E.D.

Note that for general α, β the value of S in (A2/4.11) is complex and there are <u>no</u> periodic solutions. For fixed β, there are only certain eigenvalues of α which give periodic solutions – the latter obey normal Sturm-~~Louville~~ Louville theory.

Again it is clear that Floquet's theorem applies to any second order differential equation with periodic coefficients. I am told it is used in (i) problems with elliptic boundary conditions – see my example in (b).
(ii) Motion of particles in periodic magnetic fields – this gives orbits in synchotron.
(iii) Quantum theory of electron motion in a periodic lattice i.e. under a periodic force.

(d) Eigensolutions of Mathieu's equation (optional)

This section does not have any general significance: It's most interesting feature is the use of continued fractions. As we mentioned before, in the case $\beta = 0$, Mathieu's equation becomes:

$$\frac{d^2y}{dx^2} + \alpha y = 0 \qquad \text{(A2/4.12)}$$

when the eigenvalues/solutions are:

$\alpha = 0$ $\qquad$ $y = 1$

$\alpha = 1$ $\qquad$ $y = \cos x,\ \sin x$

$\alpha = 4$ $\qquad$ $y = \cos 2x,\ \sin 2x$

$\vdots$

we find the general solutions of (A2/4.1) by expansion in powers of β. As $\cos 2x$ is even on $x \to -x$ ($x+\pi$), the two solutions retain the $x \to \pi + x$ and $x \to -x$ symmetry of their first β^0 terms. So solutions are labelled

β^0 term	full solution
1	$ce_0(x)$
$\cos x$	$ce_1(x)$
$\sin x$	$se_1(x)$
$\cos 2x$	$ce_2(x)$
$\sin 2x$	$se_2(x)$

etc.

we expand the general solution in a Fourier Series:

$$ce_{2n}(x) = \sum_k A_{2k} \cos 2kx$$

$$ce_{2n+1}(x) = \sum_k A_{2k+1} \cos(2k+1)x \qquad \text{(A2/4.13)}$$

$$se_{2n}(x) = \sum_k B_{2k} \sin 2kx$$

$$se_{2n+1}(x) = \sum_k B_{2k+1} \sin(2k+1)x$$

We just do one as an example and follow M. and W with $ce_{2n}(x)$. [Note problem 7-31 of M. and W. is $se_{2n+1}(x)$]. So recapitulate:

$$ce_{2n}(x) = A_0/2 + \sum_{m=1}^{\infty} A_{2m} \cos 2mx$$

$$\Rightarrow d^2 ce_{2n}(x)/dx^2 = \sum_{m=1}^{\infty} (-4m^2) A_{2m} \cos 2mx$$

and $[\alpha + \beta \cos 2x]\, ce_{2n}(x) = \sum_{m=1}^{\infty} A_{2m} [\alpha \cos 2mx$

$$+ \beta/2 \; (\cos 2(m+1)x + \cos 2(m-1)x)]$$

So equating coefficients of $\cos 2mx$ in (A2/4.1) to zero gives:

$$\beta A_{2m-2} + 2(\alpha - 4m^2) A_{2m} + \beta A_{2m+2} = 0 \qquad \text{(A2/4.14)}$$

a miserable 3-term recurrence relation.

Now our expression (A2/4.13) guarantees periodicity – we find the eigenvalue α in such a way that the series (A2/4.13) converges. This doesnt appear too hopeful for (A2/4.14) suggests

$$A_{2m+2} \approx \frac{8m^2}{\beta} A_{2m} \to \infty \text{ very fast}$$

like $(n!)^2$. But all is not lost, for we can rewrite (A2/4.14)

$$\beta \, A_{2m-2}/A_{2m} + 2(\alpha - 4m^2) + \beta \, A_{2m+2}/A_{2m} = 0 \qquad \text{(A2/4.15)}$$

Then if $A_{2m-2}/A_{2m} \approx 8m^2/\beta$, the singular term will be cancelled as A_{2m+2}/A_{2m} will indeed be small. Thus if A_{2m}/A_{2m-2} is small $(\beta/8m^2)$ then so will be A_{2m+2}/A_{2m}. (They're a friendly lot of coefficients). As A_{2m+2}/A_{2m} is small, it follows from (A2/4.15) that middle term dominates. A_{2m}/A_{2m-2}

i.e. $$A_{2m}/A_{2m-2} = \frac{-\beta}{2(\alpha-4m^2) + \beta A_{2m+2}/A_{2m}} \leftarrow O(\beta/m^2)$$

We can improve this by substituting for A_{2m+2}/A_{2m} to give:

$$A_{2m}/A_{2m-2} = \cfrac{-\beta}{2(\alpha-4m^2) - \cfrac{\beta^2}{2(\alpha-4(m+1)^2) - \cfrac{\beta^2}{\cdots}}} \qquad \text{(A2/4.16)}$$

This is an amazing continued fraction.

Taking $m=1$, (A2/4.16) determines A_2/A_0 starting from $\beta A_0 + 2(\alpha-4)A_2 + \beta A_4 = 0$ and succeeding equations.

Put taking $m=0$ in (A2/4.14) and supplying a special factor of 2, gives:

$$\alpha A_0 + \beta A_2 = 0. \qquad \text{(A2/4.17)}$$

Equating two expressions for A_0/A_2 i.e. (A2/4.16 and 17) gives:

$$-\frac{\alpha}{\beta} = \cfrac{-\beta}{2(\alpha-4) - \cfrac{\beta^2}{2(\alpha-16) - \dots}} \qquad \text{(A2/4.18)}$$

which is an (unusual) equation for $\alpha = \alpha(\beta)$. We can solve it by $\alpha = \alpha_0 + \alpha_0' \beta^2 + \alpha_0'' \beta^4 + \dots$ It is easy to see that the form of the continued fraction (A2/4.18) allows such an iterative determination of $\alpha_0, \alpha_0' \dots$

<u>For example</u> take $\alpha_0 = 0$.

Next approximation $\alpha = \beta^2 / 2(\alpha_0 - 4) = -\beta^2/8$

and again $\alpha = \dfrac{-\beta^2}{2(-\beta^2/8 - 4) + \beta^2/32}$

$$= \frac{-\beta^2}{8[1 + 7\beta^2/8.32]}$$

$$= -\beta^2/8 + 7\beta^4/32.8.8 + \cdots$$

and similar hard labour will give you all terms in the expansion. Note that only choice of proper zero-th order eigenvalues $\alpha = 0, 4, 16 \ldots$ give consistent solutions. (e.g. try $\alpha_0 = 1$!)

(e) General Solution of Mathieu's Equation

As we saw in (b), we also interested (for the radial equation) in non-periodic solutions. These are discussed in Abramowitz and Stegun and also M. and F. page 556 onwards. Now we showed that we always have a

Solution $f(x) = \exp(isx)\varphi(x)$: φ periodic.

M. and F. prove that s is given in terms of α and β by:

$$\sin^2[\pi s/2] = \Delta(\alpha, \beta) \qquad (A214.19)$$

This is perhaps not a completely tautological definition of Δ as Δ is a nice function and so if:

$\Delta(\alpha, \beta) < 0$ s is imaginary and solutions are ~~unstable~~ of exponential type i.e. <u>unstable</u>

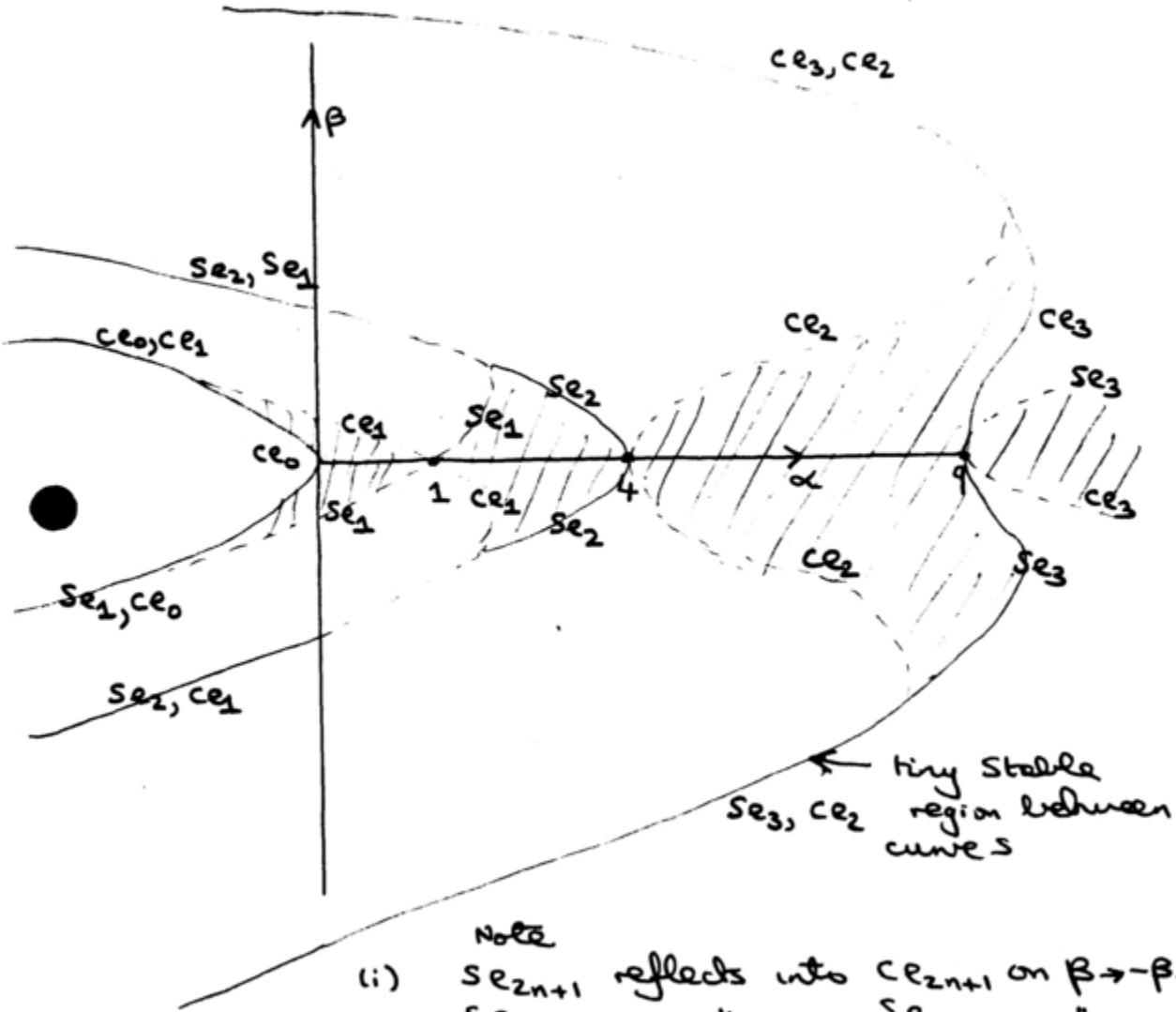
$\Delta(\alpha, \beta) = 0$ $s = 2m$: solutions are <u>periodic</u>.

$\Delta(\alpha, \beta) > 0$ and < 1 : solutions are oscillatory but <u>not</u> periodic. This is <u>stable</u> ~~[illegible]~~ region

$\Delta(\alpha, \beta) = 1$, $s = 2m+1$: solutions are <u>periodic</u>

$\Delta(\alpha, \beta) > 1$ <u>unstable</u> again

So our eigensolutions seperate regions of stable and unstable solutions.



Note

(i) se_{2n+1} reflects into ce_{2n+1} on $\beta \to -\beta$.
se_{2n} " se_{2n} "
ce_{2n} " ce_{2n} "

(ii) which regions are stable can be found from $\beta = 0$ in limit and fact that they end on eigensolutions.

A2/5 Elliptic Functions

(a) Periodic Functions of a Complex Variable

We will depart from M. and W. and follow instead the treatment of Morse and Feshbach which is more general and in many ways easier - it is certainly prettier. [M. and F. pages 425 onwards].

Consider the example of $f(x) = e^{ix}$ with x real. Then

$$f(x+2\pi) = f(x)$$
$$\text{or} \quad f(x+a) = f(x) \qquad \text{(A2/5.1)}$$

where $f(x)$ has period $a = 2\pi$.

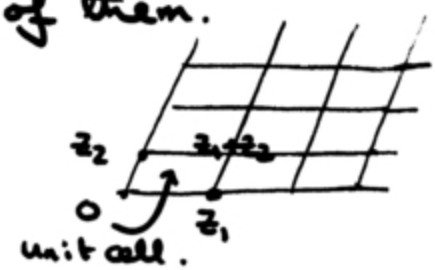
When we consider a function of two variables $[x, y = \mathrm{Re}, \mathrm{Im}\, z]$, it might be thought natural to generalize the concept of (A2/5.1) - a function defined on a 1-dimensional line - to

a function defined in two dimensions (complex plane) with two periods. [we will return to this idea when we discuss the group theory of ornaments in 1,2,3 dimensions].

$$f(z) = f(z + z_1) = f(z + z_2) \quad \text{any } z$$

and fixed z_1 and z_2. (A2/5.2)

A function satisfying (A2/5.2) which is also meromorphic (only singularities are poles) is called <u>elliptic</u>. (A2/5.2) clearly implies $f(z) = f(z + m_1 z_1 + m_2 z_2)$ for any $z, < 0$ integers m_1 and m_2. Thereby we divide the complex plane into parallelograms in which $f(z)$ repeats the values in one of them.



In accordance with the group theory example, we call the basic parallelogram the unit cell of the function. (more conventional is fundamental period parallelogram)

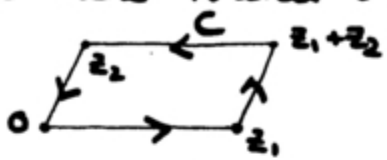
Note we can't have non-constant functions with more than two periods. [$m_1 z_1 + m_2 z_2 + m_3 z_3$ is dense].

The meromorphic functions satisfying (A2/5.2) can be elegantly classified according to their singularity (pole) structure in the unit cell. Simplest is.....

(i) If $f(z)$ has no poles in unit cell, then it is bounded as $z \to \infty$ (from periodicity) and hence must be a constant (Liouville's theorem).

(ii) more If $f(z)$ has singularities, consider $I = \frac{1}{2\pi i} \int_C f(z)\, dz$ (A2/5.3)

where the contour C runs round the unit cell.



Then from periodicity it is clear that $I=0$. But from Cauchy's theorem I is just sum of residues at poles in unit cell. Thus:

Sum of residues at poles of any elliptic function in its unit cell $=0$ (A2/5.4)

We now make a lot of money by applying (A2/5.4) to various different elliptic functions.

(iii) First note that if $f(z)$ is elliptic, so is $g(z) = f'(z)/f(z) = \partial/\partial z\,[\log(f(z))]$

So apply (A2/5.4) to $g(z)$ to find that sum of <u>it's</u> residues in unit cell are zero. Now $g(z)$ has a pole under two circumstances.

First if $f(z)$ has a pole of order m.

Then $f(z) = q(z)/(z-z_0)^m$ where $q(z)$ is regular and nonvanishing at z_0. So $\log f(z) = -m \log(z-z_0) + \log q(z)$

$$\Rightarrow \quad g(z) \approx \frac{-m}{z-z_0} + \text{regular terms}$$

i.e. $g(z)$ has a pole of residue $-m$ at z_0.

Secondly if $f(z)$ has a zero of order n, then $f(z) = q(z)\,(z-z_0)^n$ and

$$g(z) \approx \frac{n}{z-z_0} + \text{regular terms}$$

i.e. $g(z)$ has a pole of residue n at z_0.

Summing up: if $f(z)$ has poles P_i with order m_i and zeroes Z_i with order n_i, then (A2/5.4) implies

$$\sum_{P_i} m_i - \sum_{Z_i} n_i = 0$$

$$\underline{\text{or}} \quad \sum_{P_i} m_i = \sum_{Z_i} n_i \qquad \text{(A2/5.5)}$$

i.e $f(z)$ has as many zeroes as poles - counting each as many times as order.

(iv) Apply the above argument in (iii) to $g(z) = f'(z)/[f(z)-C]$ for any constant C. Then (A2/5.5) becomes

$$\sum_{p_i} m_i = \sum_{z_i^{(C)}} n_i^{(C)} \qquad \text{(A2/5.6)}$$

where $f(z)-C$ has a zero of order $n_i^{(C)}$ at $z_i^{(C)}$. Combining (A2/5.6) and (A2/5.5) we get:

An elliptic function $f(z)$ takes every value C (counted a number of times equal to order of zero $f(z)-C$) an equal number of times in each unit cell. This number is equal to the sum Σ of order of poles of $f(z)$ in unit cell. (A2/5.7)

Conformal map: As an aside, note that (A2/5.7) implies that the map $z = f(\overset{\omega}{z})$ maps the whole complex z plane Σ (Σ is defined in (A2/5.7)) times into unit cell. ~~Same~~ Alternatively we see that the elliptic function $f(\overset{\omega}{z})$ maps $1/\Sigma$ (Σ = integer remember) of z-plane into unit cell. M. and W. go through the case $\Sigma = 2$ and demonstrate the map of $\mathrm{Im}\,\overset{\omega}{z} \geq 0$ into the unit cell.

Possible Elliptic Functions

(A2/5.7) indicates the importance of Σ in classifying elliptic functions. Starting at the beginning of the world:

$\Sigma = 0$: i.e. no poles – Somewhat dull as we showed $f(z) = \text{constant}$.

$\Sigma = 1$: impossible to satisfy (A2/5.4) except with one zero residue $\Rightarrow$ constant again

$\Sigma = 2$: this is first nontrivial case and here we have two choices.

(i) One double pole : this gives Weierstrass's elliptic function treated in depth by, say, Whittaker and Watson ("A course of Modern Analysis") or Abramowitz and Stegun.

(ii) Two single poles : this is Jacobi elliptic function treated in M. and W. pages 204 → 210.

(b) Weierstrass's Elliptic Function

This can be defined by:

$$\wp(z) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{m=+\infty}{}' \left\{ \frac{1}{[z-\omega_{m,n}]^2} - \frac{1}{\omega_{m,n}^2} \right\} + \frac{1}{z^2} \qquad (A2/5.8)$$

where $\omega_{m,n} = 2m\omega_1 + 2n\omega_2$

and $'$ indicates that one should omit the term with $m=n=0$. $\wp(z)$ has periods $2\omega_1, 2\omega_2$ and is an even function of z.

We won't dwell on its properties (and its friends the θ and σ functions). However

one cannot resist the elegant theorem.

Theorem: Any elliptic function can be constructed as a rational function of $\wp(z)$ [Weierstrass's elliptic function] and a linear function of $\wp'(z)$ - its derivative. This indicates that the case $\Sigma = 2$ is in fact, essentially the general elliptic function.

Proof: To begin with, take any even elliptic function $\varphi(z)$ and let $\wp(z)$ be the Weierstrass elliptic function of the same periods. Let

$$\begin{array}{ll} a_1 \ \ldots \ a_n & \text{be zeroes of } \varphi(z). \\ -a_1 \ \ldots \ -a_n & \\ b_1 \ \ldots \ b_n & \text{be poles of } \varphi(z). \\ -b_1 \ \ldots \ -b_n & \end{array}$$

where we made explicit the fact that φ is even and put $\Sigma_\varphi = 2n$ and in list we just enter enter multiple zero/poles the number of times equal

to their multiplicity.

Now consider

$$\Phi(z) = \frac{1}{\varphi(z)} \prod_{r=1}^{n} \left\{ \frac{\wp(z) - \wp(a_r)}{\wp(z) - \wp(b_r)} \right\}$$

Now we showed in (A2/5.7) that $\wp(z)$ equals a given value, say $\wp(a_r)$ or $\wp(b_r)$ just twice. But these are precisely $z = \pm a_r$ or $\pm b_r$ and there are no more such z in unit cell. Given this we see $\Phi(z)$ has no poles for (a): zeroes of $\varphi(z)$ cancel with $\wp(z) - \wp(a_r)$ factors

(b) poles of $\wp(z)$ cancel directly for each r.

(c) zeroes of $\wp(z) - \wp(b_r)$ cancel with poles of $\varphi(z)$.

Further $\Phi(z)$ is by construction

an elliptic function. Hence as it has no singularities it must be constant (as we showed earlier).

We can now treat an arbitrary elliptic function $f(z)$ by writing:

$$f(z) = \underbrace{\tfrac{1}{2}\left[f(z) + f(-z)\right]}_{\text{even}} + \underbrace{\tfrac{1}{2}\left[f(z) - f(-z)\right]}_{\text{even}} \cdot \underbrace{\wp'(z)}_{\text{odd}}$$

and applying above argument to the two even parts exhibited.

Q.E.D.

It is clear that in the above proof we can replace $\wp(z)$ by any elliptic function that is even and has $\Sigma = 2$. Thus we can use any even Jacobi elliptic function.

(c) Jacobi Elliptic Functions.

(1) In practice these are the most often used, because of their relation to elliptic integrals. So we temporarily abandon our lofty heights and consider an elliptic integral as treated in n. and w. Consider the differential equation:

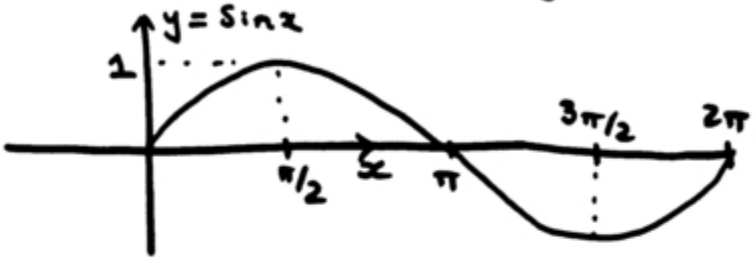
$$\left[\frac{dy}{dx}\right]^2 = (1-y^2)(1-k^2y^2) \qquad 0<k<1 \quad (A2/5.9)$$

with the boundary conditions $y=0, y'>0$ at $x=0$. A useful hint for solving this may be had by looking at the special case $k=0$ when we get solution $y=\sin x$. This shows it is better to think of y as a function of x than vice versa. Continuing the special case we can write (A2/5.9) as

$$dx = dy/\sqrt{1-y^2}$$

or $$x = \int_0^y dy/\sqrt{1-y^2} \qquad (A2/5.10)$$

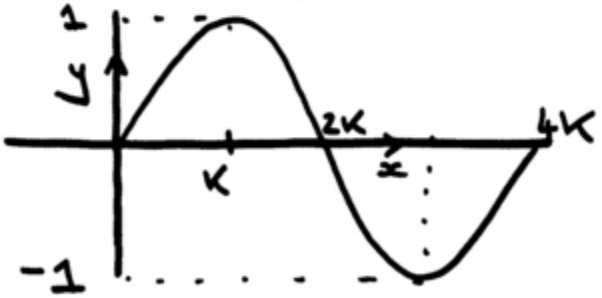
Now we know solution $y = \sin x$ is



and one can convince oneself from (A2/5.10) that this must be so. (we'll elaborate this a little later). Correspondingly, in the case $k \neq 0$ we use:

$$x = \int_0^y dy/\sqrt{(1-y^2)(1-k^2y^2)} \qquad (A2/5.11)$$

then just as before we find



y is periodic - but with a new period K given by placing $x = K$,

$y = 1$ in (A2/5.11).

i.e. $$K = \int_0^1 dy \Big/ \sqrt{(1-y^2)(1-k^2y^2)}$$

$$= \int_0^{\pi/2} d\varphi \Big/ \sqrt{1-k^2 \sin^2\varphi} \qquad \text{(A2/5.12}^1\text{)}$$

$$= \pi/2 \;\; {}_2F_1\left(\tfrac{1}{2}, \tfrac{1}{2}; 1 \,|\, k^2\right)$$

where the latter form is gotten by expanding in powers of k^2. To extend plot past $x = K$ it is neatest to go back to the D.E. (A2/5.9) and solve it with the boundary conditions $y = 1$, $y' < 0$ for $x > K$*. Then integrating (A2/5.9) replaces (A2/5.11) by:

$$x - K = \int_y^1 dy \Big/ \sqrt{(1-y^2)(1-k^2y^2)} \qquad \text{(A2/5.12}^2\text{)}$$

from whence it is obvious that

$y = 0$ at $x = 2K$ and $y = -1$ at $x = 3K$.

Writing the solution of (A2/5.9) as

* note: an easy continuity argument gives sign change in dy/dx.

$y = Sn(x)$ (Suppressing k^2 dependence)

we can thus show that Sn has periodicity p[redacted]roperty

$$Sn(x+4K) = Sn(x) \qquad (A2/5.13)$$
$$Sn(0) = Sn(2K) = 0$$

So far Sn is pretty much like Sin, But we can show that y has a second period .....

(ii) A Second Period:

Putting $y = i\bar{y}$ and $x = i\bar{x}$, (A2/5.11) becomes:

$$\bar{x} = \int_0^{\bar{y}} d\bar{y} \Big/ \sqrt{(1+\bar{y}^2)(1+k^2\bar{y}^2)} \qquad (A2/5.14)$$

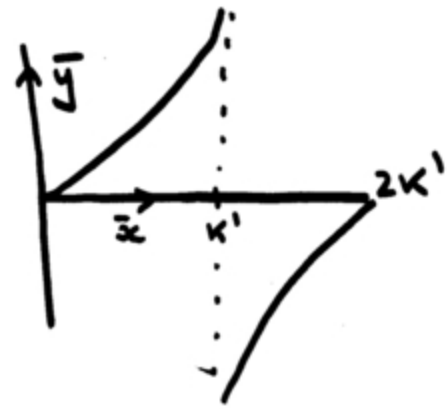
Now in the case $k=0$, this is solved by $\bar{y} = \sinh \bar{x}$ and, as we all know, $\bar{y} \to \infty$ implies $\bar{x} \to \infty$ for (A2/5.14) becomes $\int_0^{\bar{y}\to\infty} d\bar{y} \Big/ \sqrt{1+\bar{y}^2}$ which diverges to ∞ as $\bar{y} \to \infty$. For the case $k \neq 0$ however, the integral converges [redacted] and as $\bar{y} \to \infty$, $\bar{x} \to K^2$

with $K' = \int_0^\infty d\bar{y} \Big/ \sqrt{(1+\bar{y}^2)(1+k^2\bar{y}^2)}$ (A2/5.15)

Now the result $d\bar{x}/d\bar{y} \approx \frac{1}{k^2\bar{y}^2}$

implies that this singularity is a pole $\left[\therefore \bar{y} \approx \frac{-1}{k^2(\bar{x}-K')}\right]$ and

graphically



Analogously to (A2/5.12), we continue $\bar{x}$ past K' with

$$\bar{x} - K' = \int_\infty^{\bar{y}} d\bar{y} \Big/ \sqrt{(1+\bar{y}^2)(1+k^2\bar{y}^2)} \qquad \text{(A2/5.16)}$$

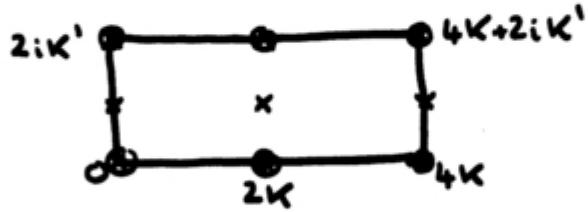
- whence $\bar{y}(2K') = 0$ and generally has period $2K'$.

Translating to $\mathrm{Sn}(x)$, we have

$$\mathrm{Sn}\,(x+2iK') = \mathrm{Sn}(x) \qquad (A2/5.17)$$

and $\mathrm{Sn}(x)$ has a pole at iK'.

Such arguments easily convince one that $\mathrm{Sn}(x)$ is an elliptic function with two poles in unit cell and hence is a Jacobi elliptic function with periods $4K$ and $2iK'$. The unit cell is shown below:



⊗ = zero
x = pole

(iii) On a technical note, put $\bar{y} = u/\sqrt{1-u^2}$ in the formula (A2/5.15) for K'. Then

$$d\bar{y} = du/\sqrt{(1-u^2)^3}$$

and
$$K' = \int_0^1 \frac{du}{[1-u^2]^{3/2}} \sqrt{\frac{1-u^2}{(1-u^2+u^2)(1-u^2+k^2u^2)}}$$

$$= \int_0^1 du/\left[\sqrt{(1-u^2)}\sqrt{1-(1-k^2)u^2}\right]$$

i.e. $$K'(k) = K\left(\sqrt{1-k^2}\right) \qquad (A2/5.18)$$

On a comparably irrelevant note, we can return to the generalized hypergeometric functions and to find that these elliptic functions can be related to basic hypergeometric functions ${}_2\Phi_1$ [page 22] with special values for the latter's arguments.

(iii) Elliptic Integrals

All this stuff is very beautiful - but is there are any reason for real live people to study it or should it be left to english clergymen. Unfortunately there is a grimy reason for studying it. Thus consider the task of evaluating

$$\int R(x, y)\, dx \qquad \text{(A2/5.19)}$$

where y is a function of x and R is any rational function of x and y.

We can consider three cases of increasing complexity:

(α) R independent of $y \equiv R(x)$. The integral (A2/5.19) is elementary.

(β) $R \equiv R(x,y)$ with y^2 a quadratic in x. The integral (A2/5.19) can be done in terms of trigonometric and hyperbolic functions; $\sin^{-1}$ and $\sinh^{-1}$.

(γ) $R \equiv R(x,y)$ with y^2 a cubic or quartic in x. These integrals can be evaluated in terms of Jacobi/Weierstrass elliptic functions. [and related theta functions]. M. and W. pages 78 to 79 or chapter 17 of Abramowitz and Stegun should be consulted for details. As an example, take

$$I = \int dx/y(x) \quad \text{where} \quad (A2/5.20)$$

$y(x)$ is a quartic. Then by a nontrivial

transformation [cf. A. and S. page 595 and 600] we can often transform variables so that y has zeroes at ± 1 and $\pm 1/k$. Then we immediately get:

$$I = \mathrm{Sn}^{-1}(x, k) = \int_0^x dx \Big/ \sqrt{(1-x^2)(1-k^2x^2)}$$

or putting $x = \sin\varphi$, we can write

$$I = \int_0^\varphi d\theta \Big/ \sqrt{1-k^2\sin^2\theta} = F(\varphi, k) \quad \sim (A2/5.21)$$

where F is an elliptic integral of the first kind. Other quartics in (A2/25.20) give a discrete set of other possibilities and other inverse elliptic ~~elt~~ functions. For instance

$$\mathrm{cn}(x) = \sqrt{1-\mathrm{Sn}^2(x)}$$

and $$\mathrm{cn}^{-1}(x) = \int_x^1 dt \Big/ \sqrt{(1-t^2)(k^2t^2+1-k^2)} \quad \sim (A2/5.22)$$

and this corresponding to the quartic with two real and two complex roots $t = \pm 1$, $t = \pm i \frac{\sqrt{1-k^2}}{k}$.

Of course when $k=0$, $cn(x) = \cos(x)$.

A3 Autonomous Systems

This is a cultural session enabling students worrying about candidacy ~~placement~~ exam to relax - for there will be no questions on it! References are Chapter 5 of Birkhoff and Rota's book we've already mentioned. This is a short account. Specialized treatises are:

R.A. Struble "Nonlinear Differential Equations" - this is on reserve in Milliken: the first chapter is very readable.

N. Minorsky "Nonlinear Oscillations".

A proper understanding of this subject will enable the reader to predict if Throop hall will fall down in a earthquake: more usefully,

given a guffer (which eats a given number of snails a day) and a fertile lot of snails, one can discover if the latter will overrun your tank or if the guffers will starve to death.

A3/1 Phase Space

(i) Consider a system of n first order ordinary D.E.'s in normal form:

$$\frac{dx_1}{dt} = X_1(x_1 \ldots x_n : t)$$
$$\vdots \qquad\qquad\qquad\qquad (A3/1.1)$$
$$\frac{dx_n}{dt} = X_n(x_1 \ldots x_n : t)$$

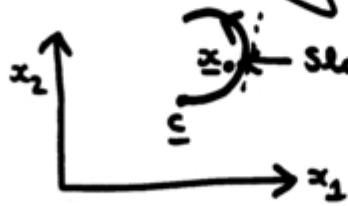
This is called <u>autonomous</u> if all the X_i are independent of t. There is however, at this stage, no other restriction on X_i: in particular they are generally <u>nonlinear</u> functions of the x_i.

Suppose god speaks to us in the night and we are given a value ~~$x_i = x$~~ $\{x_i\} = \underline{x}$ (lapsing into vector notation) at $t=0$. Call this value $\underline{x} = \underline{c}$. Then (A3/1.1) determines ~~all~~ the derivative of $\underline{x}$ at $t=0$ and hence determines a direction in space [e.g. in 2-space $dx_2/dt \div dx_1/dt$ is a direction

$\dfrac{(x_2 - c_2)}{(x_1 - c_1)} = \dfrac{dx_2/dt}{dx_1/dt}$]

Integrating wrt. t, we clearly get a curve determined by the initial value $\underline{c}$

[



Slope determined by (A3/1) for $\underline{x} = \underline{x}_0$.

]

This curve in our n dimensional space is called an <u>integral curve</u>: it is a function $\underline{x} = \underline{x}(t, \underline{c})$ $-\infty \le t \le \infty$. If t is interpreted as time and $\{x_i\}$ as

space position : then $\underline{x}(t, \underline{c})$ can be interpreted as the position of the particle at time t - where particle started off life at $t=0$ with $\underline{x} = \underline{c}$. In this interpretation (A3/1) gives the velocity of steady flow.

(ii) Example:

$$dx/dt = mx$$
$$dy/dt = m'y \qquad \text{(A3/1.2)}$$

is sort of simple and solves to

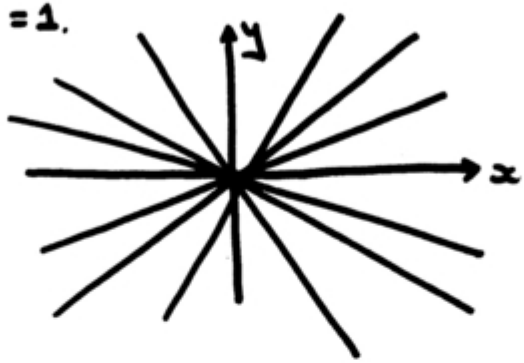
$$x = c_1 \exp(mt)$$
$$y = c_2 \exp(m't)$$

eliminating t : $y^m = k x^{m'}$ (A3/1.3)

Note we took $n=2$ in (A3/1.2) : this we will always do because $n \geq 3$ needs ≥ 3 dimensional blackboards. (So it goes).

The case $n=2$ is officially termed <u>plane</u> in the recognition of the ability of

using planar blackboards. Returning to planar example (A3/1.3) we can draw these integral curves in the case $m = m' = 1$.



when they are straight lines through the origin.

(iii) Note that many second order D.E.'s can be cast in the form of plane autonomous systems. So to prove we can do this at once for any equation whose coefficients do <u>not</u> depend on the independent variable. Thus take Newton's law $F = ma$. (I used to call it $P = mf$ but letters are

Staid in America). This is

$$\ddot{x} = F(x, \dot{x}) \qquad \text{(A3/1.4)}$$

Put $y = \dot{x}$: a standard ruse to get

$$\begin{aligned} \dot{y} &= F(x, y) \\ \dot{x} &= y \end{aligned} \qquad \text{(A3/1.5)}$$

which is, as advertised, plane autonomous. Above we defined (x, y) as "phase-space" but this is just $(x, dx/dt)$ which is normal phase space of classical mechanics. This explains the name. According to the general discussion, solutions of Newton's law are paths in phase space

we can (trivially) generalise this!

For a system with m generalized co-ordinates q_i and generalized velocities dq_i/dt

$$d^2 q_i/dt^2 = F_i[\underline{q}, dq/dt] \quad : i = 1 \ldots m \quad \text{(A3/1.6)}$$

Put $p_i = dq_i/dt$: conjugate velocity to q_i.

Then we find a 2m dimensional autonomous system.

$$\left.\begin{aligned} dq_i/dt &= p_i \\ dp_i/dt &= F_i(\underline{q}, \underline{p}) \end{aligned}\right\} \quad (A3/1.7)$$

This time we have 2m dimensional phase space spanned by p_i and q_i : $i=1..m$.

If $F(\underline{q}, \dot{\underline{q}}) = -\partial V/\partial q_i$

$V = V(\underline{q})$ only (A3/1.8)

then system is <u>conservative</u> with potential energy function V : in this case one integral of (A3/1.7) is:

$$\text{Energy } E(\underline{q}, \underline{p}) = \Sigma p_i^2/2 + V(\underline{q}) = \text{const.} \quad (A3/1.9)$$

In the $m=1$ case this is sufficient to completely determine the integral

curves as $p^2/2 + V(q) = \text{constant}$

(iv) For instance if $V(q) = q^2/2$

which corresponds to autonomous system

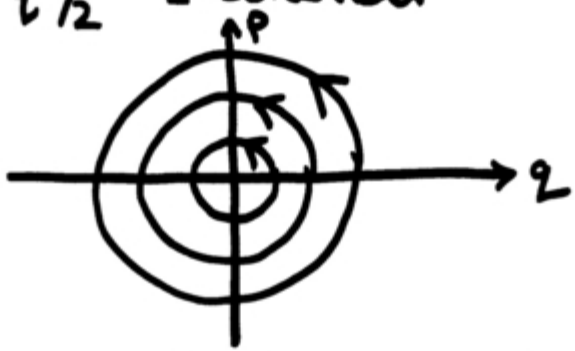
$$dq/dt = p$$
$$dp/dt = -q$$

or original second order D.E.

$$d^2q/dt^2 = -q \qquad (A3/1.10)$$

we find circular integral curves

$p^2/2 + q^2/2 = \text{constant}$



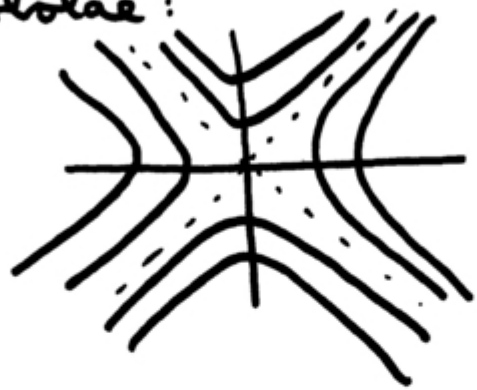
we might have made a sign error (not allowed for a Caltech graduate student of course) and put

$$V(q) = -q^2/2$$

corresponding to D.E.

$$d^2q/dt^2 = q \qquad (A3/1.11)$$

Now our integral curves are hyperbolae:



A3/2 Critical Points and Stability

(i) Now we come to our first new concept. Thus the alert reader will have noticed that the origin is a special point in the 3 phase planes drawn so far. In fact it is a critical point.

Definition: A point $\underline{x} = (x_1, \dots x_n)$ where all the functions X_i in (A3/1.1) are zero is called a critical point of the system.

Taking $n=2$ again, at a critical point $dx_1/dt = 0$ and $dx_2/dt = 0$: thus the ratio $dx_1 : dx_2$ is indeterminate. Hence (for general n) it follows that critical points are the (only) points in phase space where a unique direction and hence a unique integral curve is not defined.

In the mechanics example, critical points are :

$$p_i = 0$$

and $$dp_i/dt = \partial V/\partial q_i = 0$$

i.e. points of equilibrium.

We know from childhood examples that equilibrium points are classified by their <u>stability</u>. Thus in our examples $V(q) = q^2/2$ is stable and $V(q) = -q^2/2$ is unstable. Notice this was reflected in the different shape of integral curves for the two cases. (Stable : all integral curves are closed curves remaining near origin : Unstable : integral curves diverge from origin.)

Thus we can generalize from our intuition in this example that :

Study of integral curves near critical points $\Leftrightarrow$ stability

Quadratic terms in $V(q) \Leftrightarrow$ stability

The latter maps into:

Linear terms in $X_i(q)$ determine stability in many cases.

(ii) An example : Duffing's Equation

This is a modification of SHO equation well known to thee and me

$$d^2x/dt^2 + \omega^2 x + \beta x^3 = 0 \qquad \text{(A3/2.1)}$$

Nonlinear term

response > SHO : $\beta > 0$ hard spring

" < " : $\beta < 0$ soft spring

We get the phase space representation by the usual rule:

$$dx/dt = y \qquad \text{(A3/2.2)}$$

$$dy/dt = -\omega^2 x - \beta x^3$$

Thus: $$dy/dx = -\left[\omega^2 x + \beta x^3\right]/y$$

$$y\,dy = -\left[\omega^2 x + \beta x^3\right]dx$$

or $$y^2 + x^2\left[\omega^2 + \beta x^2/2\right] = \text{constant} = c \text{, say,} \quad (A3/2.3)$$

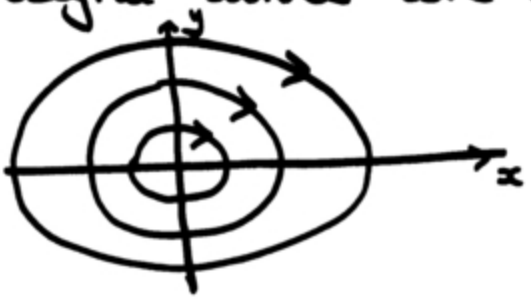
Now plugging in

$$dx/dt = y = \sqrt{c - x^2\left[\omega^2 + \beta x^2/2\right]}$$

or $$\frac{dx}{\sqrt{c - x^2(\omega^2 + \beta x^2/2)}} = dt$$

we could even integrate to find $x = x(t)$ in terms of elliptic functions. But it would be a shame to use the time we spent on the latter. Thus it is <u>not</u> necessary if we just wish to get integral curves: these are given precisely by (A3/2.3).

(a) $\beta > 0$ implies $c > 0$ in (A3/2.3) and the integral curves are easy:



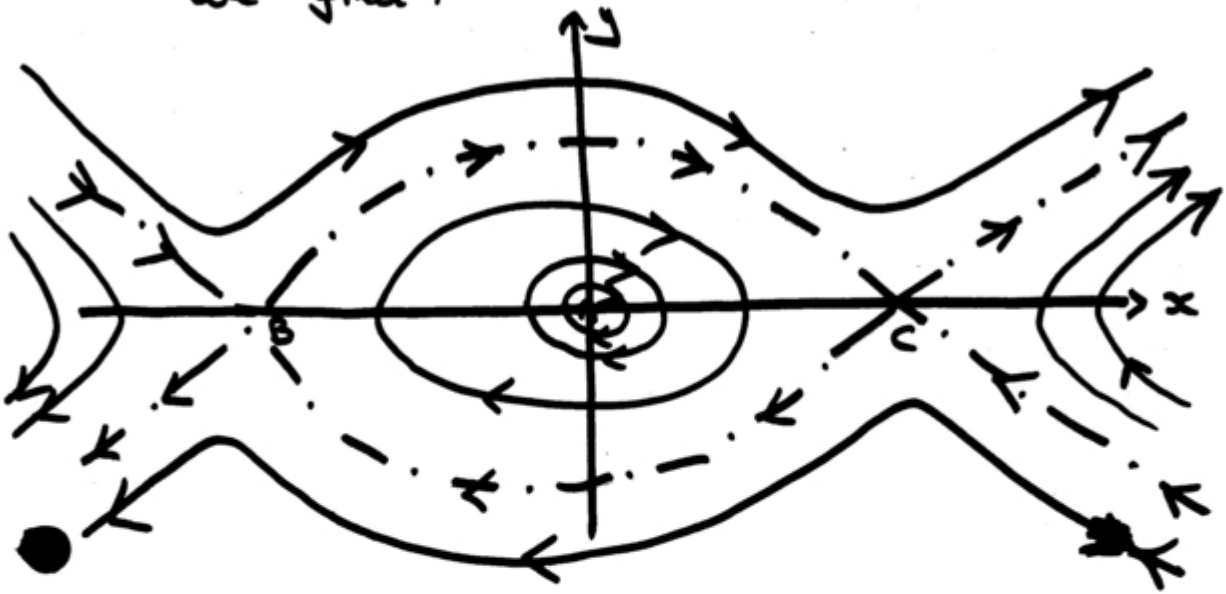
They are closed curves around the origin which is the only critical point. It is called - as we shall soon learn - a center.

(b) $\beta < 0$ is more interesting. Examination of the autonomous system reveals that the critical points are

$$y = 0$$
$$\omega^2 x = -\beta x^3$$
$$\text{i.e. } x = 0 \text{ or } x = \mp\sqrt{\omega^2/-\beta}$$

I will not go through this in detail. (See problem sheet). Letting A

be old critical point and B, C new ones, we find:



We have a much richer system. Around A - which is a STABLE CENTER - we have nice periodic solutions just as in $\beta=0$ (SHO) or $\beta>0$ case. However they are seperated by — · — curve (called a separatrix) from evil divergent solutions. The latter radiate away from B, C which are UNSTABLE saddle points

using a language we will soon come to.

Note that seperatrices and critical points are general phenomena. Recognition of when they occur plus usual graphical continuity arguments enables easy sketch of solutions.

Also note the case in which standard initial value problem i.e. given values of x and dx/dt can be solved. This gives you a point in phase space. One just locates integral curve passing through it to find ~~solut~~ qualitative nature of solution.

(iii) Stability

Let us now codify the notions on stability that have arisen in specific examples.

Definition: Let $\underline{a}$ be a critical point of the autonomous system $d\underline{x}/dt = \underline{X}(\underline{x})$: thus $\underline{X}(\underline{a}) = 0$. Then $\underline{a}$ is termed:

(i) **Stable**: if when given $\epsilon > 0$, $\exists\, \delta > 0$ such that $|\underline{x}(0) - \underline{a}| < \delta$ implies that $|\underline{x}(t) - \underline{a}| < \epsilon$ for all $t > 0$

(ii) **Asymptotically stable** when $\exists\, \delta > 0$ such that:
$|\underline{x}(0) - \underline{a}| < \delta$ implies

$$\lim_{t \to \infty} |\underline{x}(t) - a| = 0$$

(iii) **Strictly stable** when it is stable and asymptotically stable.

(iv) Neutrally Stable: when it is stable but not asymptotically stable.

(v) Unstable if it is not stable.

We will come to examples exhibiting these various concepts. Now as we indicated before the linear terms in X usually determine stability and there are some quite straightforward theorems as to when this will be so. [See B. and R. for a short account]. Thus it makes sense to study linear autonomous systems $X_i = \sum_j a_{ij} x_j$ near their critical point $\underline{x} = 0$ to understand stability. We will make our usual specialization to $n=2$ ($n \geq 3$ is again tedious but no different in principle). The main use is sketching

integral curves near critical points of general equations (these are determined by linear terms of Taylor expansion): to exhaust we only need $n=2$ for general case, we only study stability for $n=2$.

A3/3 Linear Plane Autonomous Systems

The general object under consideration is the seemingly trivial:

$$\begin{aligned} dx/dt &= ax+by \\ dy/dt &= cx+dy \end{aligned} \qquad \text{(A3/3.1)}$$

To find the associated second order D.E. (secular equation) got by eliminating y, we proceed to differentiate the first equation:

$$d^2x/dt^2 = a\,dx/dt + b\,dy/dt$$

or $d^2x/dt^2 = a\,dx/dt + b[cx + dy]$

$= a\,dx/dt + bcx + d(dx/dt - ax)$

or finally

$$d^2x/dt^2 - (a+d)\,dx/dt + (ad - bc)x = 0 \qquad \text{(A3/3.2)}$$

We are meant to know how to solve this: namely try $x = A\exp(\lambda t)$ where

$$\lambda^2 - [a+d]\lambda + [ad - bc] = 0 \qquad \text{(A3/3.3)}$$

We now get lots of different cases – depending on various types of roots of (A3/3.3). Put:

$$\Delta = [a+d]^2 - 4(ad - bc)$$

$$= (a-d)^2 + 4bc \qquad \text{(A3/3.4)}$$

So formal solution is

$$\lambda = \tfrac{1}{2}\left[(a+d) \pm \Delta^{1/2}\right] \qquad \text{(A3/3.5)}$$

We now run through cases:

(i) Focal Points : $\Delta < 0$

We get distinct complex roots in this case.

Write them $\lambda = \mu \pm i\nu$. We can then change co-ordinates (x, y), by a suitable linear transformation, such that (A3/3.1) takes for $\Delta < 0$, the canonical form.

$$\frac{dx}{dt} = \mu x - \nu y$$
$$dy/dt = \nu x + \mu y \qquad \text{(A3/3.6)}$$

Now take polar co-ordinates

$r^2 = x^2 + y^2$ gives

$$r\frac{dr}{dt} = x\frac{dx}{dt} + y\frac{dy}{dt}$$
$$= \mu x^2 - \nu xy + \nu xy + \mu y^2 = \mu r^2$$

So $r\,dr/dt = \mu r$ (A3/3.7)

Meanwhile $x = r\cos\theta$ gives

$$\frac{dx}{dt} = \frac{dr}{dt}\cos\theta - r\frac{d\theta}{dt}\sin\theta$$
$$= \mu x - y\frac{d\theta}{dt}$$
$$= \mu x - \nu y \quad \text{from (A3/3.6)}$$

So $\dfrac{d\theta}{dt} = \nu$ (A3/3.8)

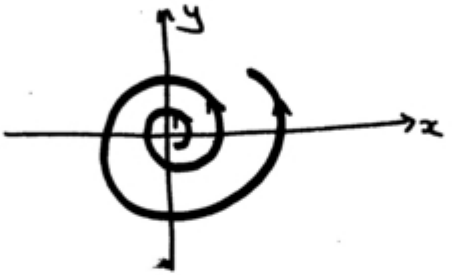
We can integrate (A3/3.7,8) to give:

$$r = \text{const} \exp(\mu t)$$

$$\theta = \nu t + \text{const}'$$

or $$r = \text{const}'' \exp(\mu\theta/\nu) \qquad \text{(A3/3.9)}$$

These are spirals:



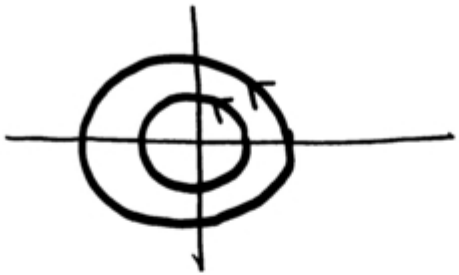
($\nu > 0$ in picture)

The arrows (direction of increasing time) are drawn for the case $\mu > 0$ when $r \to \infty$ as $t \to \infty$. Origin is in this case an UNSTABLE FOCUS.

If $\mu < 0$, then $r \to 0$ as $t \to \infty$ (reverse arrows in diagram). Now origin is a STRICTLY STABLE FOCUS.

There is a special case ~~when~~ which we've already come across in the case

of the SHO equation ($\ddot{x} + x = 0$). This is $p=0$



This time origin is a CENTRE, CENTER or VORTEX POINT. It is NEUTRALLY STABLE.

(ii) NODAL POINTS : $\Delta > 0$ and $ad > bc$

Now the roots of (A3/3.5), say $\lambda = \lambda_1, \lambda_2$, are real, distinct, and both are the same sign.

Our canonical form is

$$\frac{dx}{dt} = \lambda_1 x \quad \text{and} \quad dy/dt = \lambda_2 y \qquad \text{(A3/3.10)}$$

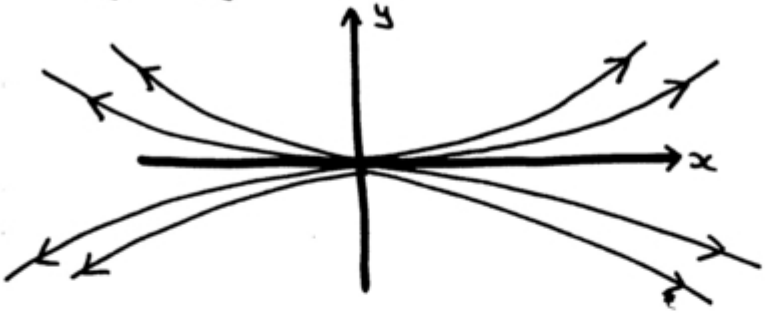
which is solved by:

$$\left.\begin{aligned} x &= a_1 \exp(\lambda_1 t) \\ y &= a_2 \exp(\lambda_2 t) \end{aligned}\right\} \qquad \text{(A3/3.11)}$$

whence $y = \text{const.}\ (\pm x)^{\lambda_2/\lambda_1}$ (A3/3.12)

To draw we assume, wolog, that $\lambda_2/\lambda_1 = |\lambda_2|/|\lambda_1| > 0$. Then take it at

use typical value $\lambda_2/\lambda_1 = 2$ to get comforting parabolas.



Drawn is $\lambda_1, \lambda_2 > 0$ which is UNSTABLE NODE.

Reverse arrows for $\lambda_1, \lambda_2 < 0$ which is a STRICTLY STABLE NODE. (Stability obvious from (A3/3.11))

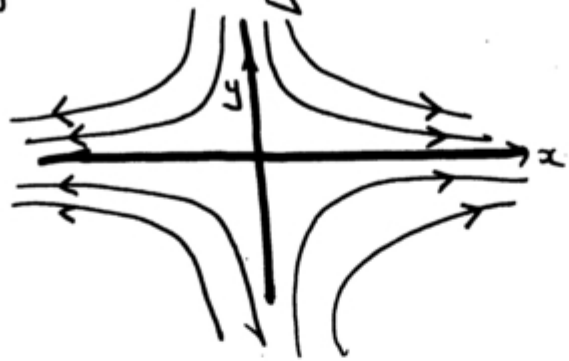
(iii) Saddle Points : $\Delta > 0$ and $ad < bc$

We now have λ = real distinct λ_1, λ_2 but they are of opposite sign. Again we choose same canonical form as in (ii) to get integral curves

$$x^{-\lambda_2/\lambda_1} y = \text{constant} \qquad (A3/3.13)$$

Now its obvious from (A3/3.11) that either x or $y \to \infty$ as $t \to \infty$ and so

origin is always unstable.



$\lambda_1 > 0$
$\lambda_2 < 0$

Origin is an UNSTABLE SADDLE POINT.

(iv) STAR $\Delta = 0$ ~~ad-bc~~ $A = \begin{bmatrix} \mu & 0 \\ 0 & \mu \end{bmatrix}$ $\therefore ad - bc = \mu^2$

we now degenerate into special cases.

If $\Delta = 0$ it is clear that

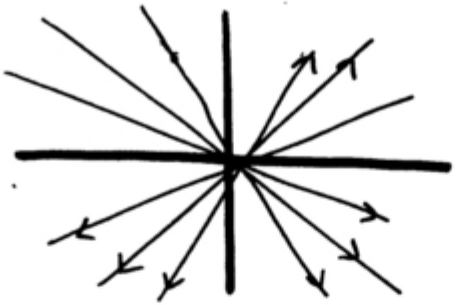
$ad - bc \geq 0$ and we take case

$ad - bc = \mu^2 > 0$.

Such a degenerate matrix can be cast into 1 of 2 standard forms. The first, treated here, is $A = \begin{bmatrix} \mu & 0 \\ 0 & \mu \end{bmatrix}$

with canonical form $\dfrac{dx}{dt} = \mu x$ (A3/3.14)

$\dfrac{dy}{dt} = \mu y$



The analysis is as in (ii). The arrows are drawn for $\mu > 0$ when the origin is an UNSTABLE STAR. For $\mu < 0$ the origin a STRICTLY STABLE STAR.

(v) NODE : $\Delta = 0$ $\quad A = \begin{bmatrix} \mu & 0 \\ 1 & \mu \end{bmatrix}$ $\quad ad - bc = \mu^2 > 0$.

This takes the second canonical form for a matrix with degenerate eigenvalues.

Thus:

$$\frac{dx}{dt} = \mu x$$
$$\frac{dy}{dt} = x + \mu y \qquad \text{(A3/3.15)}$$

Curiously this gives rise to nontrivial mathematics. Solving

$$x = c \exp(\mu t)$$
$$y = x(\text{constant} + \ln(\pm x)/\mu) \qquad \text{(A3/3.16)}$$

To draw this: note that as $x \to \infty$

$y \approx x/p \ln|x|$ which is roughly $y \approx x/p$

and as $x \to 0$, the sign of y reverses
-as $\ln|x|$ does-

$y \sim x \ln|x|/p \to 0$ as $x \to \infty$ but

$dy/dx \to \infty$ as $x \to \infty$

Drawn is $p<0$ when origin is a STRICTLY STABLE NODE. For $p>0$ origin is a UNSTABLE NODE.

(vi) Degenerate: $\Delta = 0$, $ad-bc=0$ $A = [0]$

In the case $\Delta = 0$, $ad-bc=0$ we can again have two canonical malnces. If A is identically zero we get

$$\dot{x} = \dot{y} = 0 \qquad \text{(A3/3.18)}$$

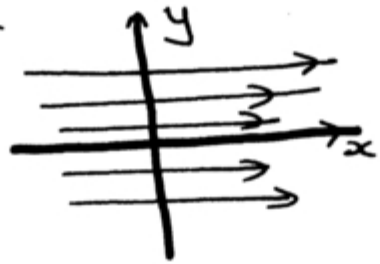
all points are Neutrally stable critical points (as ~~a~~ critical point not isolated, it is degenerate)

(vii) Degenerate : $\Delta = 0$, $ad - bc = 0$ $\quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

The other canonical form is

$$dx/dt = y \qquad \text{(A3/3.19)}$$
$$dy/dt = 0$$

whence $d^2x/dt^2 = 0$ or motion under no force



$$x = ct + c'$$

and all $y = 0$ are unstable degenerate critical points as life streams past at uniform velocity (y).

(viii) Degenerate: $\Delta \neq 0$ and $ad = bc$.

Our secular equation is

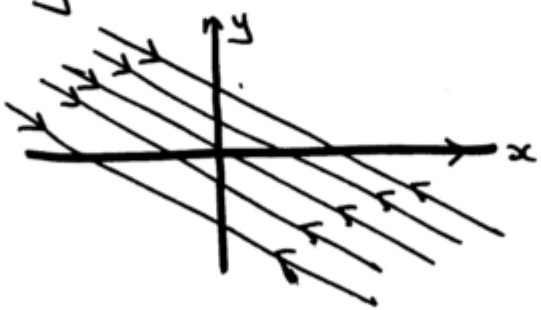
$$\ddot{x} - (a+d)\dot{x} = 0 \qquad \text{(A3/3.20)}$$

and we choose canonical form

$$\begin{aligned} dx/dt &= y \\ dy/dt &= (a+d)y \end{aligned} \qquad \text{(A3/3.21)}$$

This is uniform de-celeration under a velocity dependent damping. Integral curves are from integrating (A3/3.20) once

$$y = \text{const} + (a+d)x$$



Drawn is $(a+d) < 0$.

In this case again critical points are degenerate and consist of whole line $y = 0$. Integrating (A3/3.20) twice

$$x = c_1 + c_2 \exp((a+d)t)$$

So that if $(a+d) \gtrless 0$, critical points are unstable. If $(a+d) < 0$ critical points are stable but not asymptotically stable (given critical point Y_1, points arbitrarily near Y_1 will go to a critical point as $t \to \infty$ but it won't be Y_1 generally).

A3/4 Volterra's Equation

This is discussed in different but equivalent language in Minorsky page 65.

We have a lake or rather an aquarium with two species x and y. x (snails) feeds on vegetation and is amazingly fertile. x grows without bound:

$$dx/dt = \epsilon_1 x \qquad (A3/4.1)$$
$$\epsilon_1 > 0.$$

where we ignore possibly that vegetation runs out and there is liken unto a fine break, a snail break around our lake to ~~been~~ stop these denizens covering and devouring our green and pleasant earth.

Meanwhile we have black and yellow puffers which all other things being equal i.e. in a vacuum, gradually die:

$$dy/dt = -\epsilon_2 y \quad : \epsilon_2 > 0 \qquad \text{(A3/4.1)}$$

Now we switch up teeth of puffers and they start to eat the snails. Imagine this may be represented by the substitution in (A3/4.1,2) of:

$$\epsilon_1 \to \epsilon_1' = \epsilon_1 - \gamma_1 y$$
$$\epsilon_2 \to \epsilon_2' = \epsilon_2 - \gamma_2 x \qquad \text{(A3/4.3)}$$

This makes snails disappear in direct proportion to the number of puffers present. Correspondingly puffers appear in direct proportion to numbers of snails present. All very reasonable. So to summarize, our pond is described by:

$$\frac{dx}{dt} = \epsilon_1 x - \gamma_1 xy$$
$$\frac{dy}{dt} = -\epsilon_2 y + \gamma_2 xy \qquad \text{(A3/4.4)}$$

with ϵ_1 ϵ_2 γ_1 and $\gamma_2 > 0$. This is a plane autonomous system. Phase space is pretty nice - the axes are

our favourite observables: x and y, the number of snails and puffers. Note this equation ignores possibility of aquarium heater breaking.

It happens that one can integrate (A3/4.4) to give

$$F(x,y) = \exp\left[-\gamma_2 x - \gamma_1 y\right] x^{\epsilon_2} y^{\epsilon_1} = \text{const.} \qquad \text{(A3/4.5)}$$

This can be checked easily: differentiate the above wrt t once:

$$-\gamma_2 F\, dx/dt - \gamma_1 F\, dy/dt + \epsilon_2 \frac{F}{x}\, dx/dt + \epsilon_1 \frac{F}{y}\, dy/dt = 0$$

or $$(\epsilon_2 y - \gamma_2 xy)\, dx/dt + (\epsilon_1 x - \gamma_1 xy)\, dy/dt = 0 \qquad \text{(A3/4.6)}$$

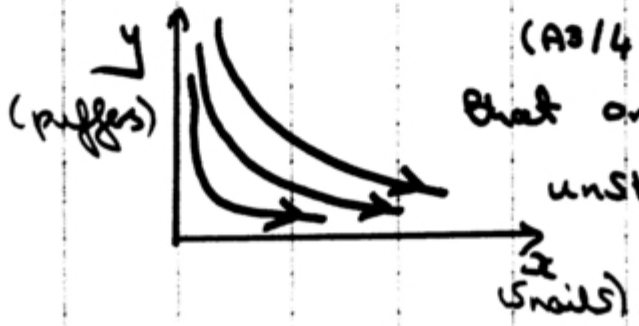
of course t is not defined by (A3/4.5), <u>choosing</u> it to satisfy first part of (A3/4.4), then implies from (A3/4.6) the second part. Thus (A3/4.5) is the true equation for integral curves of (A3/4.4).

However let's ignore this accident of solubility. Rather see how we can sketch the integral curves without solving the equation. The idea is to find the special points: sketch the solutions near them – then use continuity.

The first critical point is the origin. In accordance with our previous discussion, linearize equation to find behaviour near there:

i.e. $$\begin{aligned} dx/dt &= \epsilon_1 x \\ dy/dt &= -\epsilon_2 y \end{aligned} \qquad (A3/4.7)$$

and we are back to the toothless puffer situation.



(A3/4.7) shows that origin is an unstable saddle point.

But there is another critical point:

$$\epsilon_1 - \gamma_1 y = 0$$
$$-\epsilon_2 + \gamma_2 x = 0$$

Let the solution be $x = (x_1, y_1)$.

Note this is physical i.e x_1, y_1 both >0!

Put $\hat{x} = x - x_1$

$\hat{y} = y - y_1$

$$d\hat{x}/dt = \epsilon_1 x_1 - \underset{0}{\underset{\shortparallel}{\gamma_1 x_1 y_1}} + \hat{x}(\underset{0}{\underset{\shortparallel}{\epsilon_1 - \gamma_1 y_1}})$$

$$- \gamma_1 x_1 \hat{y} - \gamma_1 \hat{x} \hat{y}$$

or linearizing $\frac{d\hat{x}}{dt} = -\gamma_1 x_1 \hat{y}$ (A3/4.8)

Similarly $\frac{d\hat{y}}{dt} = \gamma_2 y_1 \hat{x}$

To find nature of critical points we compare with

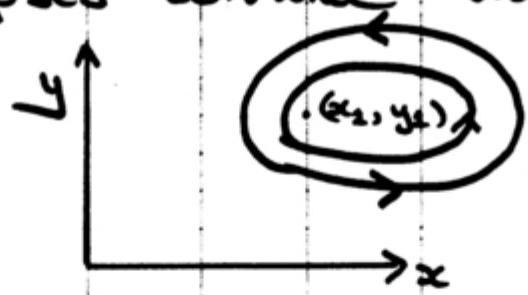
$$d\hat{x}/dt = a\hat{x} + b\hat{y}$$
$$d\hat{y}/dt = c\hat{x} + d\hat{y}$$

and examine roots of

$$\lambda^2 - [a+d]\lambda + [ad-bc] = 0$$

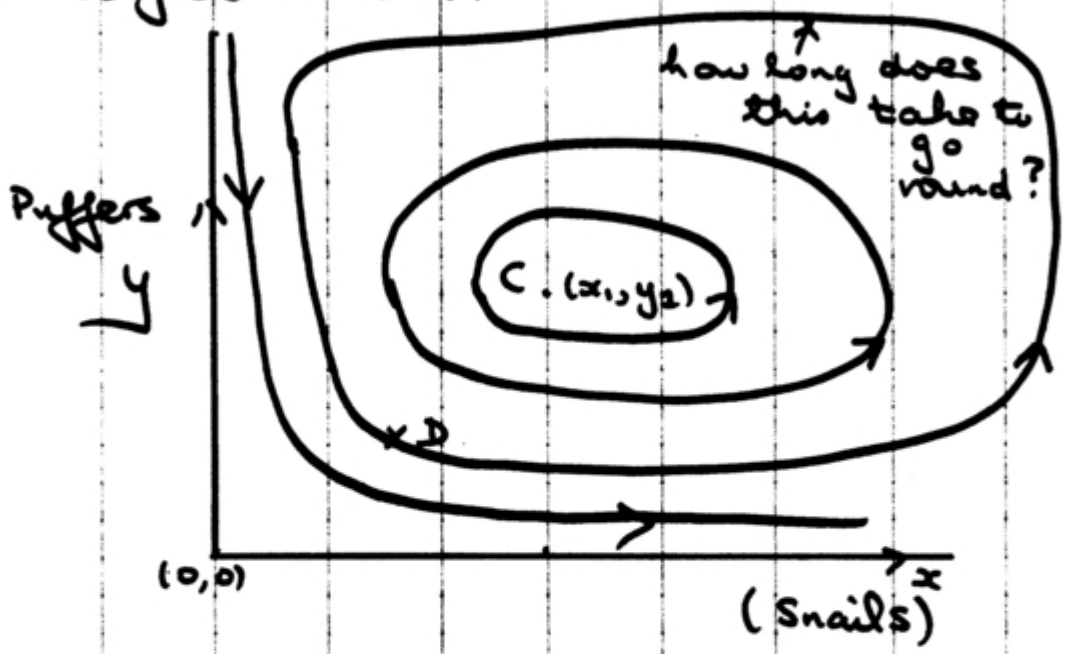
i.e. $\lambda^2 - 0 \cdot \lambda + \gamma_1 \gamma_2 x_1 y_1 = 0$

i.e. λ has equal and opposite pure imaginary roots. This corresponds to a center in previous discussion: the integral curves are ellipses around this critical point



arrows are t increasing and direction from examination of (A3/4.8)

Putting the behaviour near the two critical points together we get as integral curves:



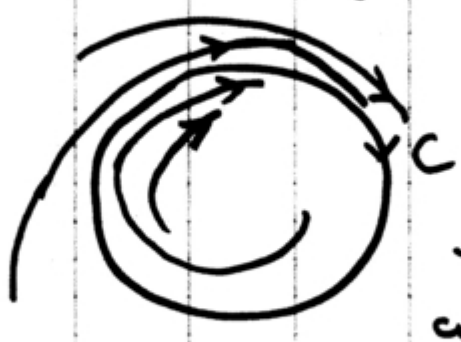
All of them are closed. C is an equilibrium point. Curves give fluctuations about this. The curiosity is that if you start at D, one gets periodic fluctuations of extremely large amplitude. There is no damping. This seems unphysical. More reasonable would be a limit cycle - which we now study.

A3/5 Limit Cycles of Poincaré

(i) This is discussed in B. and R. page 142.

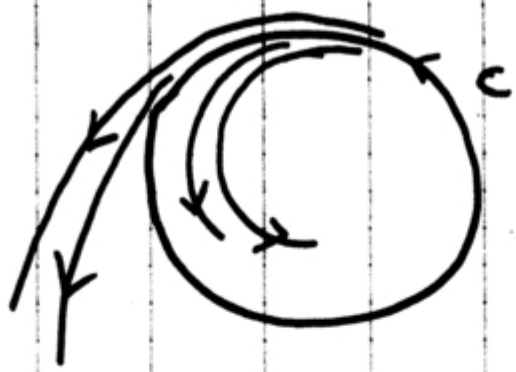
So far we have only studied stable or unstable points, i.e. critical points, which are the limits of neighbouring integral curves. (if stable). An interesting feature of many nonlinear D.E.'s is the existence of stable or unstable closed curves which are limits of neighbouring ~~closed~~ integral curves.

Pictorially



C is a stable limit cycle

alternatively



C is a unstable limit cycle

One can also have limit cycles that are stable from one side and unstable from the other.

(ii) Lets look at the following example

$$dx/dt = y + \frac{x}{\sqrt{x^2+y^2}}\left[1-(x^2+y^2)\right]$$

$$dy/dt = -x + \frac{y}{\sqrt{x^2+y^2}}\left[1-(x^2+y^2)\right]$$

which is certainly a nonlinear plane autonomous system. Turns out that all known soluble examples of limit cycles correspond to trivial equations in polar co-ordinates which can become

quite neatly looking when transformed to x, y space. This is no exception. So put:

$$x = r\cos\theta \qquad y = r\sin\theta \qquad \text{(A3/5.1)}$$

Then $r^2 = x^2 + y^2$ gives

$$r\,dr/dt = x\,dx/dt + y\,dy/dt$$

$$= xy + x^2/r\,[1-r^2]$$
$$-yx + y^2/r\,[1-r^2]$$

<u>or</u> $$dr/dt = 1 - r^2$$

again we can prove

$$d\theta/dt = -1 \qquad \text{(A3/5.2)}$$

Integrating $\theta = -t + \text{constant}$

and $$r = \frac{Ae^{2t} - 1}{Ae^{2t} + 1} \qquad \text{(A3/5.3)}$$

putting $t = 0$, $r = r_0$ gives

$$A = \frac{1 + r_0}{1 - r_0}$$

For convenience, we allow $-\infty \le r \le \infty$, rather than usual $0 \le r \le \infty$.

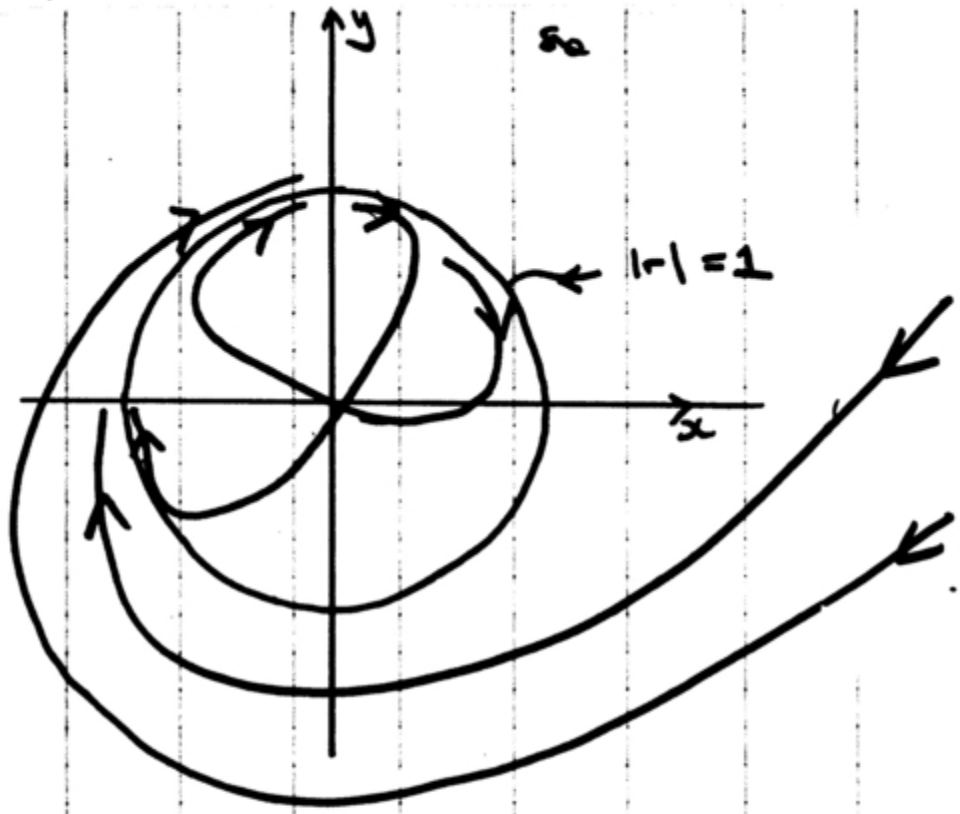
Then if $r_0 < 1$, A is > 1 and as $t \to \infty$, $r \to 1$ from below as $t \to -\infty$, $r \to -1$ from above

Again if $r_0 > 1$, A is < -1
$t \to +\infty$, $r \to 1$ from above
to examine negative t, write:

$$r = \frac{|A|\, e^{2t} + 1}{|A|\, e^{2t} - 1} \qquad : r_0 > 1$$

i.e. $r \to \infty$ as $t \to -\frac{1}{2} \log|A| < 0$
This occurs along a given ray as $\theta \approx -t + \text{constant}$

$|r| = 1$ is therefore a stable limit cycle.



So

note $d\theta/dt < 0$ and this determines direction (negative r doesnt affect this). All curves go through origin. Note latter is rather odd as in limit $x, y \to 0$, X_1 and X_2 (in language of (A3/1.1) become singular due to $1/\sqrt{x^2+y^2}$ factor

A3/6 Van der Pol's Equation

(i) This is

$$d^2y/dt^2 - \mu(1-y^2)\,dy/dt + y = 0 \qquad \text{(A3/6.1)}$$

This is a simple example of an important class of equations. It is a normal oscillator equation with a damping term $-\mu(1-y^2)\,dy/dt$ which changes sign at $y=1$.

$\mu > 0$, $y < 1$ it is negatively damped as dy/dt term gives opposite sign to y for d^2y/dt^2. Thus oscillations tend to increase in size. For $y>1$ it reverses its effect and oscillations tend to be reduced in sign. This change in sign of amplitude of oscillation gives hope of a limit cycle around about $y=1$.

We use the customary dodge to convert (A3/6.1) into an autonomous system i.e.

$$p = dy/dt$$

$$dp/dt = \mu(1-y^2)p - y \qquad \text{(A3/6.2)}$$

First we examine for critical points. The only one is $y = p = 0$. Linearizing

$$dy/dt = p$$

$$dp/dt = \mu p - y \qquad \text{(A3/6.3)}$$

which are the exact autonomous system corresponding to the ($\mu > 0$) negatively damped oscillator.

$$d^2y/dt^2 - \mu\, dy/dt + y = 0 \qquad \text{(A3/6.4)}$$

From above argument we would expect origin to be <u>unstable</u>. This is the case, for the associated

eigenvalue equation $\det(A - \lambda I) = 0$

has $A = \begin{bmatrix} 0 & 1 \\ -1 & \mu \end{bmatrix}$

with eigenvalues given by

$$\lambda(\lambda - \mu) + 1 = 0 \quad \text{or} \quad \lambda = \tfrac{1}{2}\left(-\mu \pm \sqrt{\mu^2 - 4}\right) \qquad (A3/6.5)$$

Examining our previous results we see origin is unstable focus ($|\mu| < 2$).

(arrows from fact that unstable and for $y = 0$, dy/dt same sign as p).

For the simple equation (A3/6.4), this is the complete solutions and foci diverge to infinity. However in Van der Pol's case, the damping changes sign for large y^2 and so stops divergence and gives limit cycle.

Lets show this in detail for the limiting case of small μ. Go into our favourite polar co-ordinates. (suggested because for $\mu = 0$, integral curves are circles $y^2 + p^2 = \text{constant}$)

$$\rho = y^2 + p^2$$

$$\varphi = \tan^{-1}(p/y) \qquad \text{(A3/6.6)}$$

Then (A3/6.2) becomes :

$$d\rho/dt = 2\mu\,[1 - \rho\cos^2\varphi]\,\rho\sin^2\varphi$$

$$d\varphi/dt = -1 + \mu\,[1 - \rho\cos^2\varphi]\,\sin\varphi\cos\varphi \qquad \text{(A3/6.7)}$$

as direct evaluation readily reveals.

Now yet again for small ρ we see that $d\rho/dt > 0$ and ρ grows.

In the small μ limit, $\varphi \approx -t$ and hence $d\rho/dt \approx -d\rho/d\varphi$. Then

~~it clears that for large ρ~~

the average over an orbit = average over t is approximately equal to average over φ. Hence on average for large ρ, $1-\rho\cos^2\varphi$ is <0 and size of orbit tends to decrease. Once again we must have a limit cycle between these two limits. We can find the limit cycle by noting that (a) in limit of $\mu \to 0$ it must become a circle (b) it clearly must satisfy

$$\int_{\text{orbit}} d\varphi \; d\rho/d\varphi = 0 . \qquad \text{(A3/6.8)}$$

ρ is now just a number and combining (A3/6.8) and (A3/6.7) gives

$$\int_0^{2\pi} \left[\sin^2\varphi - \rho\cos^2\varphi\sin^2\varphi\right] d\varphi = 0 \qquad \text{(A3/6.9)}$$

or $\int_0^{2\pi} \left[\tfrac{1}{2}(1-\cos 2\psi) - \rho/4 \left(1-\cos^2 2\psi\right)\right] d\psi = 0$

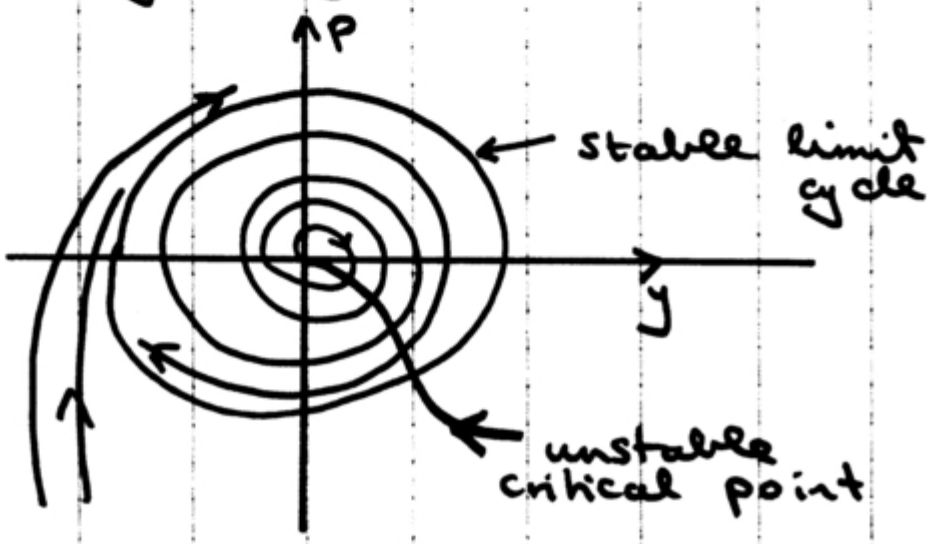
or $\int_0^{2\pi} \left[\tfrac{1}{2}(1-\cos 2\psi) - \rho/4 \left(1-\tfrac{1}{2}(1+\cos 4\psi)\right)\right] d\psi = 0$

or $2\pi \left[\tfrac{1}{2} - \rho/8 \right] = 0$

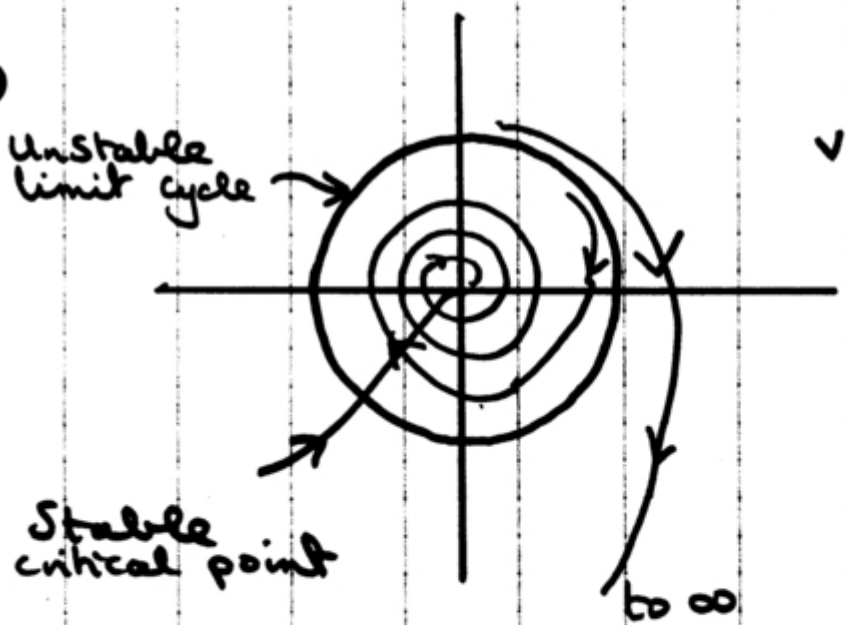
i.e. $\underline{\underline{\rho = 4}}$

and for small μ system has limit cycle

$$y^2 + (dy/dt)^2 = 4 \qquad \text{(A3/6.10)}$$



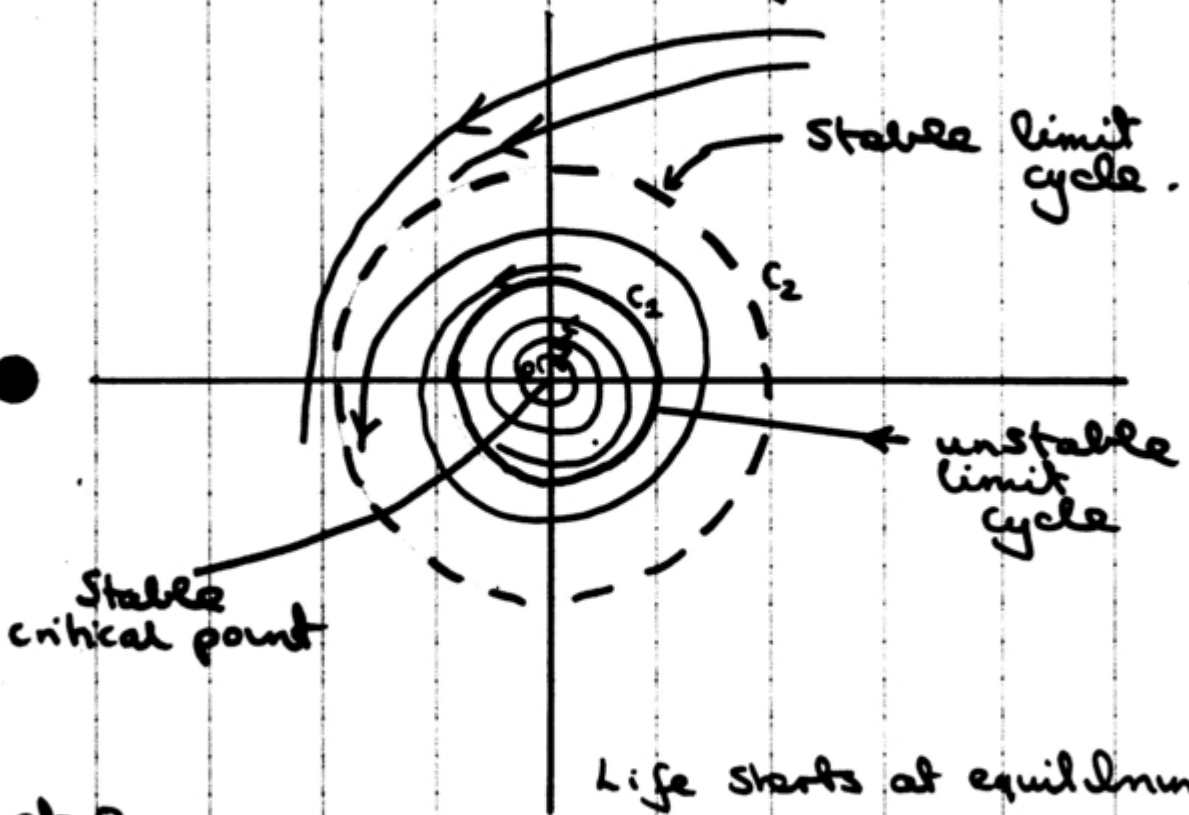
This situation is termed "Soft self excitation". Any nonzero impulse will take system from its unstable equilibrium point at the origin to the stable orbit that is the limit cycle. Note how stability of critical points/limit cycles alternates. As an example take $\mu < 0$ in (A3/6.1)



$\mu < 0$ Van der Pol:

(ii) A real physical situation will have a more complicated set of critical points and limit cycles.

Here is "hard self-excitation":



Life starts at equilibrium at O.

If impulse not enough to get past C_1, system will decay back to stable rest. However if it gets past C_2 its stable mode will be nontrivial oscillation C_2.

(iii) One can prove that limit cycles exist for certain classes of D.E's. For instance Van der Pol's equation is an example of one satisfying the following conditions.

Take Lienards equation

$$\ddot{y} + f(y)\dot{y} + q(y) = 0 \qquad (A3/6.11)$$

Let $q(y) = y\,h(y)$: $h(y) > 0$

Let $f(y)$ be < 0 in some interval $[a,b]$ of y including the origin, and > 0 outside this interval.

Let $q(y) = -q(-y)$

$f(y) = f(-y)$

and $\int_0^\infty f(y) = +\infty$

Given all this we have <u>Levinson-Smith theorem</u> that (A3/6.11) has a unique limit cycle to which every nontrivial integral curve tends. See Minorsky page 203 or B. and R. page 144 for

details of proof.

"Non-trivial" in above is meant to exclude in, say, Van der Pols equation the point at the origin which is an isolated "integral curve".

A3/7 Conclusions

(i) When we try to plot a graph $y = f(x)$, the standard technique is to plot special values and limits and use continuity to join up results. One lesson of previous sections is that even if we can't solve equation (as was case for Van der Pol), graphical techniques can be used. Again we locate special "points" - here critical points and limit cycles - and use continuity to draw general integral curves.

(ii) When can we use phase space technique? This is discussed in Strutt chapters 2 and 3.

We clearly need (for the continuity etc.) that there is a unique trajectory through every point. (or rather "general" point - it wasn't

true at some critical points). This implies that initial value problem - given x, dx/dt for $t = t_0$ in plane case - is always soluble and has a unique solution. This is not obvious: thus

$$dx/dt = x^{1/2} \qquad (A3/7.1)$$

with $x = 0$ at $t = 0$ has two solutions

$$x \equiv 0$$

$$\text{and } x = t^2/4 \qquad (A3/7.2)$$

However there are general criteria: take autonomous equations in vector form

$$d\underline{x}/dt = \underline{X}(\underline{x}) \qquad (A3/7.3)$$

for vectors $\underline{x}$ and $\underline{X}$ of dimension n. Suppose $\underline{X}$ satisfies Lipschitz condition i.e. $|\underline{X}(\underline{x}) - \underline{X}(\underline{y})| \leq M|\underline{x} - \underline{y}|$ for some constant M where $\underline{x}$ and $\underline{y}$ run over some region R. Then one can prove that:

(a) Initial value Problem Soluble

(b) Solution is unique

(c) Answer is a continuous function of t. [this is ~~clear~~ obvious for (A3/7.3) states that derivative w.r.t. t of $\underline{x}$ exists].

(iii) Finally - on a different tack - that we have not discussed "time" to traverse a given integral curve. This is often important e.g. δ is sometimes infinite!

A4 The WKB method

This is a method associated with quantum-mechanical applications, which is however interesting simply from a mathematical point of view as a perturbative approach to an otherwise insoluble problem. It is treated in many places: we follow M. and W. Section 1-4 (pages 27 onwards) and M. and F. page 1092 to 1105. We consider the differential equation:

$$y'' + f(x)y = 0 \qquad \text{(A4.1)}$$

Note any second order linear homogeneous (i.e. no y-independent term) linear D.E. may be transformed into this form. Thus if y satisfies the general $y'' = p(x)y' + q(x)y$ and we put $y(x) = Y(x)\exp\{\tfrac{1}{2}\int p(x)dx\}$, then $Y(x)$ satisfies an equation of the desired form (A4.1).

Now we are interested in solving (A4.1) when $f(x)$ is slowly varying – the latter in a sense soon to be defined in (A4.5). Of course if $f(x)$ is constant $= f$ say, then the solutions to (A4.1) take the familiar form $\exp[\pm i\sqrt{f}\,.x]$. It is thereby deemed natural to change dependent variables from y to φ where

$$y = \exp[i\varphi(x)] \qquad \text{(A4.2)}$$

Then

$$y' = i\varphi' y$$

$$y'' = i\varphi'' y + i\varphi' y' = [i\varphi'' - (\varphi')^2]\, y$$

and (A4.1) becomes:

$$-(\varphi')^2 + i\varphi'' + f = 0 \qquad \text{(A4.3)}$$

We solve this approximately – neglecting φ'' completely in (A4.3). We get $\varphi' = \pm\sqrt{f}$ and

$$\varphi(x) = \pm \int \sqrt{f(x)}\, dx \qquad \text{(A4.4)}$$

$$= \text{const} + x\sqrt{f}$$ in the special case $f(x) = \text{const}(f)$. (A4.4) says that in

place of the constant $\sqrt{f}$, one takes for slowly varying $\sqrt{f(x)}$, the "path average"

$$\int \sqrt{f(x)}\, dx \Big/ \int dx .$$

We can now check the consistency of our assumption (neglecting φ'') for using (A4.4) to estimate:

$$\varphi'' \simeq \pm \tfrac{1}{2} f'/\sqrt{f} \qquad \text{(A4.5)}$$

converts the condition $|\varphi''| \ll |f|$ into

$$\tfrac{1}{2} |f'| \ll |f|^{3/2} \qquad \text{(A4.6)}$$

This defines the condition on f of "slow variation" which is necessary for the validity of the method. We can improve the estimate (A4.4) by using not $\varphi''=0$ in (A4.3) but rather (A4.5). This gives:

$$(\varphi')^2 = f \pm i/2\, f'/\sqrt{f}$$

$$\text{or } \varphi' = \pm\sqrt{f} + i/4\, f'/f$$

$$\text{whence } \varphi(x) = \pm \int \sqrt{f(x)}\, dx + i \ln f/4 \qquad \text{(A4.7)}$$

or with $y = \exp[i\varphi(x)]$, the general solution of (A4.1) is:

$$y(x) = \frac{1}{[f(x)]^{1/4}} \left\{ c_+ \exp\left[i \int \sqrt{f(x)}\, dx \right] + c_- \exp\left[-i \int \sqrt{f(x)}\, dx \right] \right\} \qquad (A4.8)$$

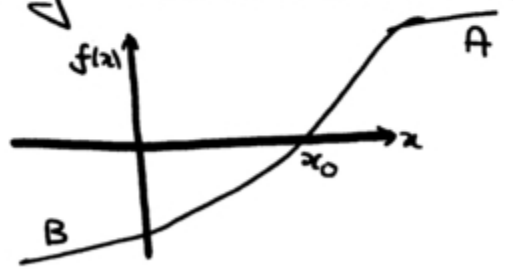
This is the WKB approximation. Why stop here you may ask: the extra $1/[f(x)]^{1/4}$ between (A4.8) and (A4.4) for $y(x)$ seems quite important. This is a legitimate question: the error in (A4.8) may be investigated in two ways (i) continue iterative process, substituting (A4.7) for φ'' in (A4.3). This gives an expansion for φ' in terms the small parameter $i f'/f^{3/2}$ and its derivatives with <u>real</u> coefficients so

$$\varphi' = \pm\sqrt{f}\left(1 + O\left(f'^2/f^3\right)\right) + i/4\, f'/f \left(\quad " \quad " \quad \right) \qquad (A4.9)$$

(ii) Alternatively we can follow the

Greens function (M. and F. page 1046 bottom) or (equivalently) variation of parameters (M. and W. equation (1-102)) method to derive an exact expression for the error in (A4.8). We will return to this after (A4.1). The expansion (A4.9) will certainly break down when $f=0$ and indeed $y(x)$ given by (A4.8) becomes quite silly under these circumstances. We can overcome this by two methods.

(i) M. and W. (having set the problems) know that one is mostly interested in a problem sketched here when we have one zero



at x_0 but otherwise $f'/f^{3/2}$ is small. Then the WKB method works well at A

or B and the only problem is connecting solutions in these two regions i.e. problem is relating $c_{\pm}^{A}$ to $c_{\pm}^{B}$ and one isn't too interested in solution near x_0. Thus M. and W. ingeniously solve this by connecting A and B by a complex path which is chosen to avoid difficult places where $f'/f^{3/2}$ is large.

(ii) M. and F. however bravely find a solution valid throughout full region and we follow this method because although one must accept more miracles and analytic results as faith, the argument is clearer. The rudiments of this approach are given in M. and W., section 4-5 and problem 7-36. Note that if we write $f(x) = g(x)(x-x_0)$ then for large

$(x-x_0)$ slowly varying f implies slowly varying g and in particular, for constant $g(x)$, f does satisfy (A4.6). So it is natural - just as (A4.2 to 8) was inspired by the case of solution to $y'' + fy = 0$ (f const) - to consider solutions of

$$d^2y/dx^2 + g(x-x_0)\,y = 0 \qquad \text{(A4.10)}$$

with constant g. This is Airy's equation

We now present the first act of faith as (A4.10) is solved by $y = \psi_\pm(x)$ where

$$\psi_+(x) = \sqrt{x-x_0}\; J_{1/3}\left[\tfrac{2}{3} g^{1/2}(x-x_0)^{3/2}\right]$$
$$\psi_-(x) = \quad " \quad J_{-1/3}\left[\quad " \quad\right] \qquad \text{(A4.11)}$$

Now we must invent a miracle, for define $w(x) = \int_{x_0}^{x} \sqrt{f(x)}\,dx$ (A4.12),

then clearly as $x \to x_0$,

$$w(x) \approx \tfrac{2}{3} g^{1/2}(x-x_0)^{3/2} \qquad \text{(A4.13)}$$

Further the behaviour as $\omega \to +\infty$ (real) of $J_\nu(\omega)$ is $\sqrt{\frac{2}{\pi\omega}} \cos\left[\omega - \tfrac{1}{2}\pi(\nu + \tfrac{1}{2})\right]$

It follows that the function

$$P = \frac{\sqrt{\omega}}{f^{1/4}} \left[A\, J_{1/3}(\omega) + B\, J_{-1/3}(\omega)\right] \quad (A4.14)$$

approaches a linear combination of $\psi_\pm$ as $\omega \to 0$ ($x \to x_0$) and the WKB approximation (A4.8) as $\omega \to \infty$. Thus it is a candidate for an interpolating formula valid through the zero (x_0) and in the asymptotic regions A and B of our figure. It is by no means clear (to me) that one has proved (as one has, say, for (A4.8)) that the error in P does not "build up" between x_0 and the asymptotic regions A, B. Rather this must be checked — unfortunately the same method used in

(A4.2); i.e put $y(x) = Y[\varphi(x)]$ where Y satisfies $Y'' + xY = 0$, appears clumsy and so the logical development of the subject suffers another blow as we follow M. and F. and note that P satisfies

$$\frac{d^2P}{dx^2} + \left[f(x) - \frac{r''(x)}{r(x)} \right] P = 0 \qquad \text{(A4.15)}$$

where $r(x) = w(x)^{1/6} / f(x)^{1/4}$

So $r''(x)/r(x)$ is error which is finite for $x = x_0$ but small being proportional to $f''(x_0)$: asymptotically $x \to \pm\infty$ the error tends to zero. The "build-up" can be studied by writing (A4.15) as

$$LP = 0$$

whence (A4.1) is

$$(L + r''/r)\, y = 0$$

Letting the error $\epsilon = y - P$, we get

$$L\epsilon = -r''y/r \qquad \text{(A4.16)}$$
$$= -r''P/r - r''\epsilon/r$$

This is can be converted into an integral equation for ϵ using Green's function methods. This can be solved by standard techniques. Alternately we can use variation of parameters as do M. and W. in their analogous discussion of the case with no $(x-x_0)$ factor (equations (1-93) (1-102)).

On a final formal note we note that case of two (close) zeros of $f(x)$ can be discussed similarly in terms of the

solutions of $y'' + \underset{\text{const.}}{q}(x-x_0)(x-x_1)y = 0$. This is described in M. and F. pages 1103 to 1105.

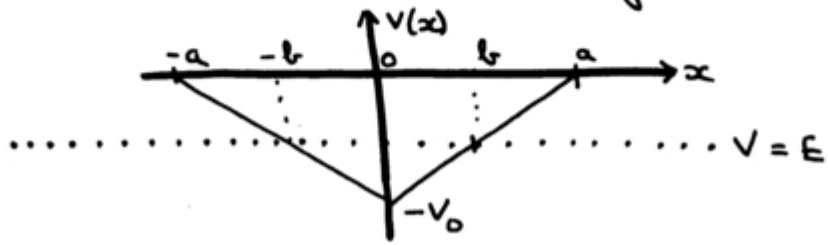
The case of two far-away zeroes can be treated as in the following examples

As an example of the utility of the WKB method, let us do problem 1-39 of M. and W.

Here we wish to solve

$$y'' + [E - V(x)]\, y = 0 \qquad \text{(A4.17)}$$

where $V(x)$ is the triangle:



in the bound state region $0 < E < -V_0$. One can of course solve this in terms of "elementary functions" but it would be tedious to match these at the discontinuities of the potential. The net value of $f(x) = E - V(x)$ is < 0 for $|x| > b$ and so the solutions are exponential. The boundary conditions imply that we want the "negative" solution which $\to 0$ as $x \to \pm\infty$.

Consider the region $0 \le x \le \infty$ including one zero at $x = b$. It is convenient to use in (A4.14) an argument $\omega = \int_x^b \sqrt{f}\, dx$ which is well defined for $x < b$ when $\sqrt{f} > 0$. Taking $x > b$ we have $\sqrt{f} = \exp(i\pi/2)\sqrt{-f}$ and as $\omega \propto (b-x)^{3/2}$, $\omega = \exp(3i\pi/2)\, W$ with W real > 0. Now we must look up the relevant asymptotic behaviour of $J_\nu(\omega)$ to find that for such a phase $(3\pi/2)$

$$J_\nu(\omega) \to e^{i\pi(\nu+1/2)} \sqrt{\frac{2}{\pi z}} \cos\left(z + \tfrac{1}{2}\pi(\nu+\tfrac{1}{2})\right) \qquad (A4.18)$$

So
$$J_{1/3}(\omega) \to \frac{1}{2}\sqrt{\frac{2}{\pi\omega}} \left[\exp[-W + i5\pi/12] + \exp[W + i5\pi/4] \right]$$
and
$$J_{-1/3}(\omega) \to \frac{1}{2}\sqrt{\frac{2}{\pi\omega}} \left[\exp[-W + i\pi/12] + \exp[W + i\pi/4] \right] \qquad (A4.19)$$

Inserting (A4.19) into (A4.14), we see that

we achieve the promised exp[-W] behaviour only if A=B. We can now forget $x > b$, and just insert A=B in the asymptotic expansion of (A4.14) for $w = \int_x^b \sqrt{f}\, dx$ real >0. This was given just above (A4.14) and we get:

$$P \rightarrow \frac{c_1}{f^{1/4}} \cos\left[\int_x^b \sqrt{f(t)}\, dt - \pi/4 \right] \quad (A4.20)$$

. We can similarly analyse the region $-\infty \leq x \leq 0$ with one zero at $-b$ to find for $x > -b$.

$$P \rightarrow c_2/f^{1/4} \cos\left[\int_{-b}^x \sqrt{f(t)}\, dt - \pi/4 \right] \quad (A4.21)$$

We must match the two expressions (A4.20), (A4.21) which letting arguments of cos be θ_1, θ_2 respectively implies $\theta_1 + \theta_2 = n\pi$: n integer. only this will give an x-independent condition which is

$$\int_{-b}^{b} \sqrt{E - V(x)}\, dx = (n + 1/2)\pi$$

or $$\frac{4}{3} a \frac{(V_0 + E)^{3/2}}{V_0} = (n + 1/2)\pi \qquad (A4.22)$$

The condition (A4.6) for the validity of the method gives

$$V_0/a \ll (V_0 + E)^{3/2} \qquad (A4.23)$$

i.e combining (A4.22), (A4.23), the WKB method is valid in classical limit of large n.