EcosimPro integration wrt a space variable instead of TIME

Sommaire

1 Purpose

In some cases, one needs to solve a differential equation with respect a variable that is not the TIME.
EcosimPro can solve only wrt the TIME.
So one can use many other tools to do that, or still using our favourite EcosimPro but considering the TIME as a space variable.
This is performed here below.

2 Problem

The shape of the free surface is provided by the equality of the hydrostatic forces and the capillary forces.
Equations are provided below, involving the radius of curvature of the free surface, so the first and second derivatives of the free surface curve.

♦ Phénomènes d'ascension capillaire près d'une paroi

La longueur capillaire l_c intervient dans de nombreux phénomènes. Elle représente par exemple l'ordre de grandeur de la hauteur d'ascension capillaire d'un fluide le long d'une paroi verticale dans le cas où le fluide mouille celle-ci (à la différence de la figure 1.17), c'est-à-dire lorsque le centre de courbure de l'interface est à l'extérieur du liquide (Fig. 1.19).

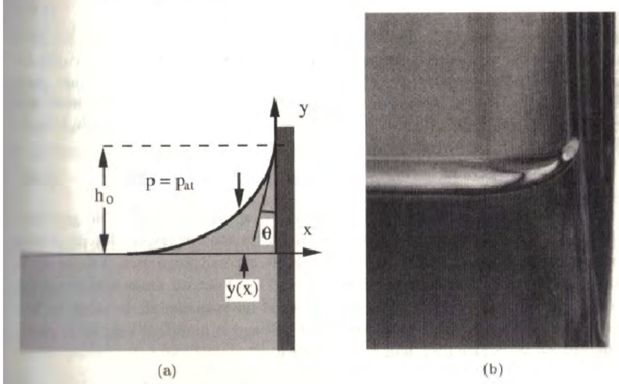


Fig. 1.19 – *Ascension d'un fluide mouillant sur une paroi verticale : (a) schéma ; (b) expérience réalisée avec de l'eau dans un récipient en verre. La hauteur d'ascension est de l'ordre du millimètre (photographie C. Rousselin, Palais de la Découverte).*

La pression à l'intérieur du liquide à une ordonnée $y\,(x)$ au-dessous de l'interface s'écrit : $p_s(x) = p_{at} - (\gamma/R(x)) + \rho\, g\, y\,(x)$ où $R(x)$ représente le rayon de courbure local de l'interface (il est positif dans le cas présent où le centre de courbure est au-dessus de la surface), p_{at} est la pression de l'air au-dessus de l'interface, le terme en γ/R représente les effets de la tension superficielle et le terme $\rho\, g\, y\,(x)$ ceux de pression hydrostatique. D'autre part, juste au-dessous de l'interface dans sa partie plane, la pression dans le liquide est p_{at}. On en déduit l'égalité :

$$\rho\, g\, y\,(x) = \frac{\gamma}{R(x)} \cdot \tag{1.62}$$

En utilisant la relation géométrique $R(x) = [(1 + y'(x)^2)^{3/2}]/y''(x)$, on obtient l'équation différentielle :

$$\rho\, g\, y = \gamma \frac{y''}{(1+y'^2)^{3/2}} \tag{1.63}$$

qui peut se mettre sous la forme :

$$\mathrm{d}(y^2) = -2\frac{\gamma}{\rho\, g}\,\mathrm{d}\left(\frac{1}{\sqrt{1+y'^2}}\right) \cdot \tag{1.64}$$

On voit apparaître l'échelle de longueur du problème qui est la longueur capillaire $l_c = \sqrt{\gamma/\rho\, g}$. Par intégration, en utilisant les conditions asymptotiques pour x très grand (en valeur absolue), $y \to 0$ et $y' \to 0$, et en remarquant que $y'(x=0) = -\cot\theta_0$, on obtient pour la hauteur d'ascension capillaire le long de la paroi :

$$h_0^2 = 2 l_c^2 (1 - \sin\theta_0). \tag{1.65}$$

La hauteur d'ascension le long de la paroi est donc bien de l'ordre de grandeur de la longueur capillaire, corrigée par un facteur faisant intervenir les propriétés de mouillage de la paroi par le liquide (angle de raccordement θ_0). Le cas de la figure 1.19 est celui d'un fluide mouillant ($\theta_0 < 90°$). Pour un fluide non mouillant comme le mercure ($\theta_0 > 90°$), l'interface descend au voisinage de la paroi au lieu de remonter, et h_0 représente la hauteur dont descend le liquide le long de la paroi, au-dessous de sa surface libre loin de cette paroi.

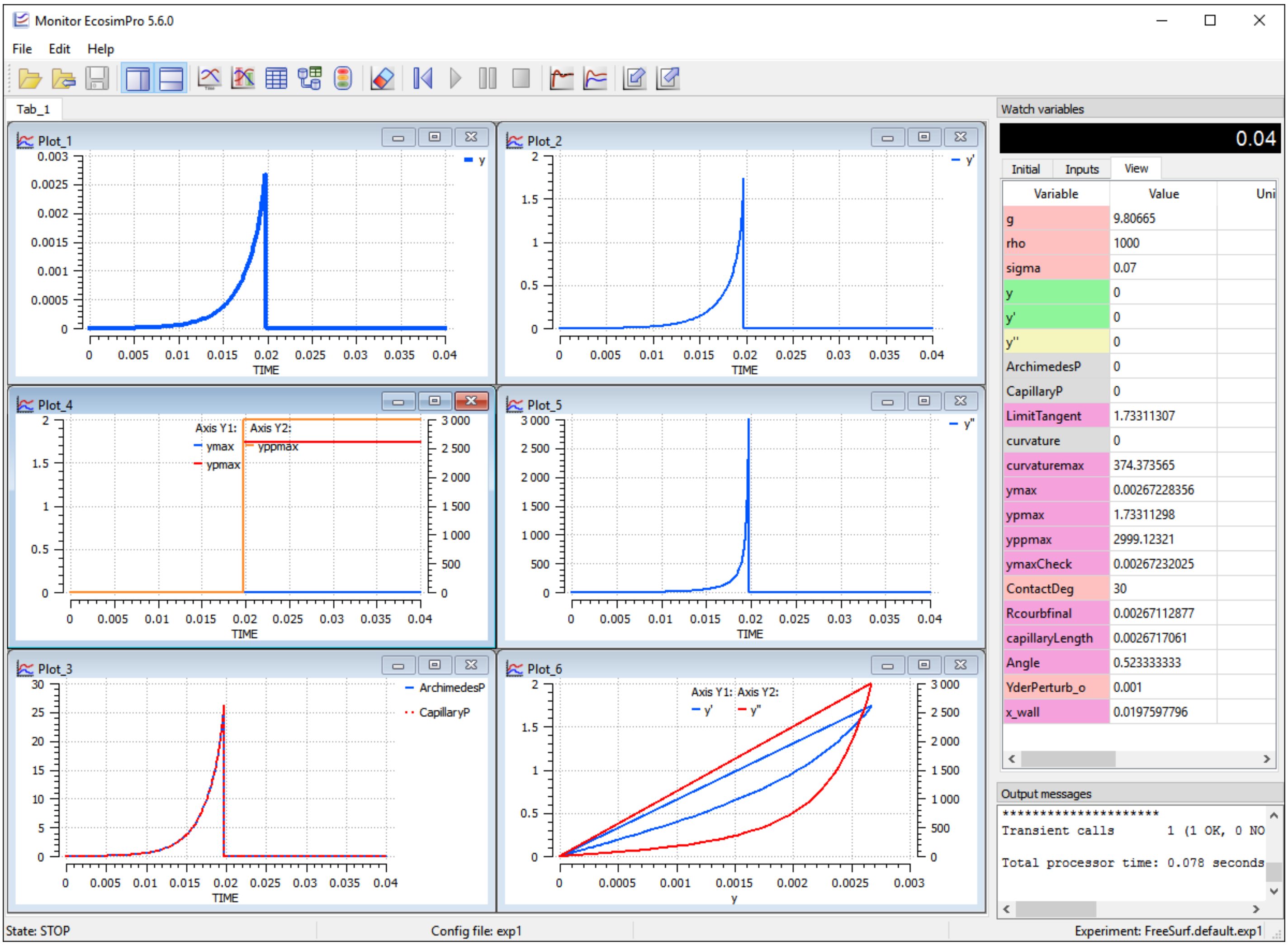
3 Results

One start the integration at x=0 (i.e. TIME=0). The integration of the curve *y(x)* is finished when x reaches the wall : that is where (or when) the condition on the tangent to the wall is satisfied ($y'(x)=\cos\theta/\sin\theta$). This give a position *x_wall* given by the TIME at which this occurs.

But all depends on the initial conditions of the problem.

With null values for y and for y' one stay on a straight line $y=0$ whatever x is (i.e. TIME)

So with non null y' the process run visibly perfectly (the final value of the height at the wall is very near to the theoretical value given in the reference).



4 Conclusions

It seems to work perfectly...

But why is it so perfect?
What is the rule for being able to do so?
Here the problem shows that y and y'' are proportional: does this allow to solve in this manner?
Any other reasons?

No other conclusions for now, many questions to solve!

5 Tracability listing

```
/*-----------------------------------------------------------------------------------
 LIBRARY: MY_TOOLS
 FILE: FreeSurface
 AUTHOR: Koppel
 COMPANY: KopooS
 DESCRIPTION: diff equation
 CREATION DATE: 29/04/2017
-----------------------------------------------------------------------------------*/
-- ' 30/04/2017 17:59:41
COMPONENT FreeSurf
	-- equa diff wrt to a spatial variable x, so this is not wrt time, but one use time as it was equal to x
	-- example on how to solve non-time dependant equation while using the time as an equivalent spatial variable
	DATA
		REAL rho=1000
		REAL g= 9.80665
		REAL sigma=20e-3 --N/m
		REAL ContactDeg=20  -- angle free surface wrt a wall
		REAL YderPerturb_o=1e-10 --initial perturbation of the first derivative needed to start the not-null integration
	DECLS
		REAL curvature
		REAL y -- y represent the free surface height (m) with respect to a spatial variable x (= time, so  1 s = 1 m )
		REAL ArchimedesP, CapillaryP
		DISCR REAL LimitTangent --when the fluid reach the vertical wall, the shape of y is imposed by the contact angle, which is a tangent of y  =dy/dx = cos(c)/sin(c)
		DISCR REAL Angle, x_wall, ymax, ypmax,yppmax,curvaturemax, Rcourbfinal --saved values
		--check wrt real solution
		DISCR REAL ymaxCheck, capillaryLength

	INIT --at initial time=0
		y=0
		y'=YderPerturb_o -- a small perturbation on the tangent (first derivative)   y'= dy(x)/dx   is given in order to allow the integration which will be stopped by a WHEN directive
		Angle=ContactDeg*3.14/180 --in radian
		LimitTangent=cos(Angle)/sin(Angle)                          --y''=1e-10
	--curvature'=1   not good
		capillaryLength=sqrt(sigma/(rho*g))
		ymaxCheck=sqrt( 2*capillaryLength**2*(1-sin(Angle))) --according to "Hydrodynamique Physique Etienne Guyon et al, CNRS éditions, pp60 ed.2001

	DISCRETE
		WHEN (y'==LimitTangent) THEN -- termination normal of the integration process
			--save the current values when this case happen
			x_wall=TIME
			ymax=y
			ypmax=y'
			yppmax=y"
			curvaturemax=curvature
			Rcourbfinal=1/curvature
			--Froze the further integration
			y'=0
			--also needed to froze the integration process from now on directly from y itself
			y=0
		END WHEN

	CONTINUOUS
		curvature=y"/(1+y'**2)**1.5
		ArchimedesP=rho*g*y
		CapillaryP=sigma*curvature  -- radius of curvature not suited  when y" is zero Rc=infinite division by zero  so use curvature= 1/Rc
		ArchimedesP=CapillaryP --               the free surface is solving this equation    (
	y=sigma/(rho*g)*y"/(1+y'**2)**1.5   for very low y': y=sigma/(rho*g)*y''*(1-1.5*y'**2)   y and y" are proportional    )

END COMPONENT
```

```
/*-----------------------------------------------------------------------------------
 LIBRARY: MY_TOOLS
 COMPONENT: FreeSurf
 PARTITION: default
 EXPERIMENT: exp1
 TEMPLATE: TRANSIENT
 CREATION DATE: 29/04/2017
-----------------------------------------------------------------------------------*/

-- ' 30/04/2017 18:00:56
EXPERIMENT exp1 ON FreeSurf.default
  DECLS
  OBJECTS
  INIT
	-- initial values for state variables
    BOUNDS
  BODY
	-- creates an ASCII file with the results in table format
	-- REPORT_TABLE("results.rpt", "*")
	-- set the debug level (valid range [0,4])
	DEBUG_LEVEL= 1
	-- select default integration solver. Valid methods are IDAS (_SPARSE), DASSL(_SPARSE), CVODE_BDF(_SPARSE), CVODE_AM, RK4, EULER, AM1, AM2 and AM4
	IMETHOD= IDAS -- default is DASSL, recommended is either IDAS or IDAS_SPARSE
	-- Settings for different actions. Valid actions are: SEV_DISABLE, SEV_NONE, SEV_WARNING, SEV_ERROR, SEV_KILLPOINT, SEV_FATAL
	eSetErrorAction(ERR_BAD_OPER,SEV_KILLPOINT) -- Detect bad numerical operations (eg division by zero), default is SEV_NONE, recommended is SEV_KILLPOINT
	eSetErrorAction(ERR_NAN_INF, SEV_KILLPOINT) -- Detect NaN or Inf values, default is SEV_NONE, recommended is SEV_KILLPOINT
	eSetConfig(CFG_FORCE_STOP_CINT,TRUE) -- Force or not to stop the solver each CINT. Sometimes you can speed up the simulation if you select FALSE
	-- set tolerances and other important inputs
	REL_ERROR = 1e-06 -- transient solver relative tolerance
	ABS_ERROR = 1e-06 -- transient solver absolute tolerance
	TOLERANCE = 1e-06 -- steady solver relative tolerance
	INIT_INTEG_STEP =-1 -- initial integration step size (-1 means use the solver estimation)
	MAX_INTEG_STEP =-1 -- maximum integration step size (-1 means use the solver estimation)
	NSTEPS = 1 -- Only for explicit solvers use CINT/NSTEPS as integration step size
	REPORT_MODE = IS_STEP--EVENT -- by default it reports results at every CINT and event detection. Other valid options are IS_STEP, IS_CINT and IS_MANUAL_REFRESH
    YderPerturb_o=1e-3 --1e-13 --1e-1--1e-10
    ContactDeg=30
    sigma=70e-3 --20e-3 --N/m
	-- simulate a transient in range[TIME,TSTOP] reporting every CINT
	TIME = 0
	TSTOP =.04 -- 15
	CINT = 0.001
	INTEG()
END EXPERIMENT
```
