UCLA, Department of Mathematics
Ecole Polytechnique

-

Rapport d'option

-

DIFFUSION GENERATED MOTION BY MEAN CURVATURE

-

Pierre Mascarenhas*
promotion 89

-

April-July 1992

*Research supported in part by the Army Research Office grant #DAAL03-91-G-0162

1 Introduction

There are several ways of tracking a front propagating under certain conditions. One of them consists in replacing the front by a finite number of points, which will then be moved according to the front's propagation law. This method is not always very simple to handle, especially when parts of the front merge together, when there are sharp angles or when there are more than two regions.

For general propagation laws and for two regions motion there exists another method, based on a Hamilton-Jacobi level set formulation[1]. If this propagation law is an affine function of the curvature , then a very simple way of tracking the front exists. This motion can be computed using a diffusion kernel.

[1] see *Front Propagating with Curvature dependent Speed: Algorithms Based on Hamilton Jacobi Formulations* by Stanley Osher and James A. Sethian , Journal of Computational Physics, V 79, (1988)

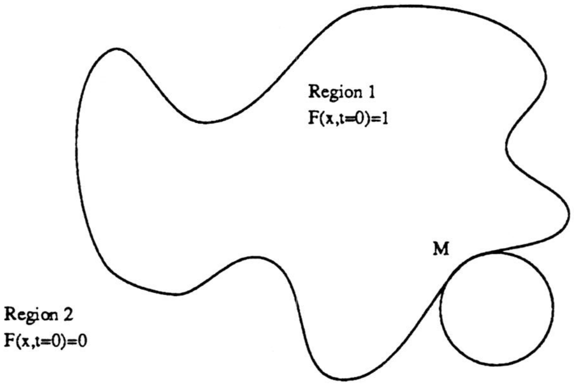
2 Diffusion Generated Motion

The idea of this method[2] is quickly motivated by looking at the diffusion kernel written in polar coordinates. It reads:

$$\frac{\partial F}{\partial t} = \frac{D}{R}\frac{\partial F}{\partial r} + D\frac{\partial^2 F}{\partial r^2} + \frac{D}{R^2}\frac{\partial^2 F}{\partial \theta^2}$$

for a two dimensional problem.

A front usually splits the plane in two regions, at least locally (we will not yet consider more complex topologies, such as triple points). In the following F(X,t=0) will be the characteristic function of one of those two regions. It is equivalent to chose F as the characteristic function of Region1 or Region2, because $F1 + F2 = 1$ for $t = 0$, and this property remains true for $t > 0$ if we diffuse $F1$ and $F2$. The level set 1/2 of both functions will be equal.



If we write the above 2-dimensional equation in coordinates centered at the center of curvature of a given point, M, of the front, we have initially around M:

$$\frac{\partial^2 F}{\partial \theta^2} = 0$$

.

[2] see *Diffusion Generated Motion by Mean Curvature* by Barry Merriman, James Bence, Stanley Osher, UCLA CAM report 92 - 18 (1992)

the equation then reduces to:

$$\frac{\partial F}{\partial t} = \frac{D}{R}\frac{\partial F}{\partial r} + D\frac{\partial^2 F}{\partial r^2}$$

which is clearly the addition of motion at speed $\frac{D}{R}$ and diffusion. The level set $F(x,t) = 1/2$ will have an initial velocity of $\frac{D}{R}$. If we take the level set $F(x,dt) = 1/2$ as being the new front with a small enough time step and iterate this operation we will obtain a curvature dependent motion at speed $\frac{D}{R}$, where R is the local radius of curvature.

For an n-dimensional problem there is not center of curvature valid for all directions but the following calculation (and proof) yields a speed of $(n-1)D$ times the sum of the inverses of the principal radii of curvature in all directions.

3 Proof of the convergence of the algorithm

In this section we will prove that the level $F = 1/2$ mentioned above has the correct initial propagation speed, and that therefore the algorithm converges toward the curvature dependent motion described above when the time step is sufficiently small. We will use the explicit solution to the heat-operator:

$$\frac{\partial F}{\partial t} = D\Delta F$$

which yields

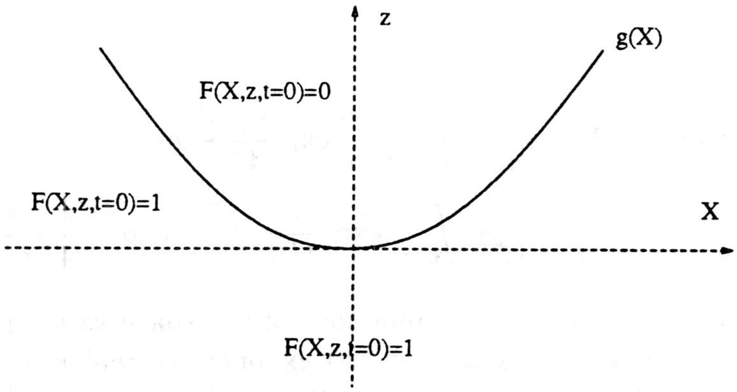
$$F(X,t) = \frac{1}{(4\pi Dt)^{n/2}} \int_{\Re^n} \exp \frac{-(X-Y)^2}{4Dt} F(Y,0)\partial Y$$

Those functions are not differentiable when t=0 so we can not use results about implicit functions, and have to calculate approximations.

We will now suppose that the boundary of the expanding area is smooth, at least 3 times differentiable and with a bounded third derivative.

We chose a system of coordinates centered on a point of the expanding front and suppose it is tangent to z=0, where z is the last coordinate.

The initial front can then be written at least locally as a smooth function, g(X), of the $n-1$ first coordinates, $x^1, \ldots, x^{n-1}$. We will suppose that it is the case globally, and then add the corresponding correction terms. The capital letters will designate a vector relative to the $n-1$ first coordinates.

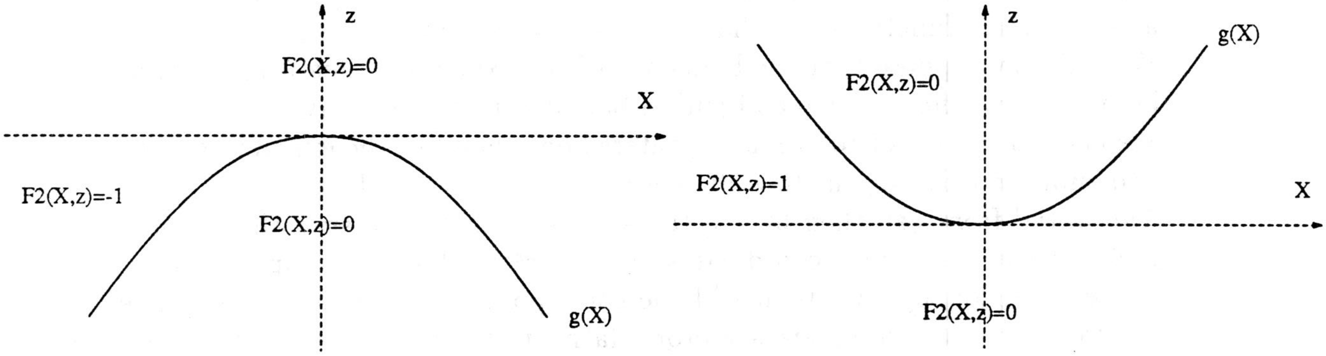


We have:

$$F(0,z,t)=\frac{1}{(4\pi Dt)^{n/2}}\int_{\Re*\Re^{n-1}}\exp\frac{-((z-y)^2+X^2)}{4Dt}F(X,y,0)\partial y\partial X$$

$$\begin{aligned}F(0,z,t) &= \frac{1}{(4\pi Dt)^{1/2}}\int_{\Re}\exp\frac{-(z-y)^2}{4Dt}H(y)\partial y\\ &+ \frac{1}{(4\pi Dt)^{n/2}}\int_{\Re*\Re^{n-1}}\exp\frac{-((z-y)^2+X^2)}{4Dt}F_2(X,y)\partial y\partial X\end{aligned}$$

where H is the characteristic function of the negative numbers, and $F_2=1$ if $0<y<g(X)$, $F_2=-1$ if $0>y>g(X)$, 0 otherwise. (the sum of $F_2(X,y)$ and $H(y)$ gives back the original F).



case g concave

case g convex

Then:

$$\begin{aligned}F(0,z,t) &= 1/2-\frac{1}{(4\pi Dt)^{1/2}}\int_0^z\exp\frac{-(y)^2}{4Dt}\partial y\\ &+ \frac{1}{(4\pi Dt)^{n/2}}\int_{\Re^{n-1}}\exp\frac{-X^2}{4Dt}\int_0^{g(X)}\exp\frac{-(z-y)^2}{4Dt}\partial y\partial X\end{aligned}$$

We will now define k_i bounded functions of the space variables and t. We then approximate and integrate the first exponential, replace g by its second order approximation, and approximate the third exponential. Then:

$$F(0,z,t) \;=\; 1/2 - \frac{z}{(4\pi Dt)^{1/2}} + (z/\sqrt{t})^3 k_1 + I(z,t) \;\;,$$

where I is :

$$\begin{aligned} I(z,t) \;=\; & \frac{1}{(4\pi Dt)^{n/2}} \int_{\Re^{n-1}} \exp\frac{-X^2}{4Dt} \\ & \int_0^{\sum_{i=1}^{n-1}\frac{a_i}{2}x_i^2 + k_3\|X\|^3} (1 + \frac{z^2+y^2}{t}k_2)\partial y \partial X \;\;, \end{aligned}$$

and where the a_i are the second derivatives of g in its principal directions. (We suppose that in our coordinate system the Hessian of g is already diagonal).
We then have:

$$\begin{aligned} I(z,t) \;=\; & \frac{1}{(4\pi Dt)^{n/2}} \int_{\Re^{n-1}} \exp\frac{-X^2}{4Dt} \\ & (\sum_{i=1}^{n-1}\frac{a_i}{2}x_i^2 + k_3\|X\|^3 + \frac{\|X\|^6 P_1(X)}{t} + \frac{z^2\|X\|^2 P_2(X)}{t})\partial X \;\;, \end{aligned}$$

P_1 and P_2 being bounded by polynomials in X.
This leads to (P'_1 and P'_1 are also bounded by polynomials):

$$I(z,t) \;=\; \frac{(4Dt)^{3/2}\pi^{1/2}}{2(4\pi Dt)} \sum_{i=1}^{n-1} \frac{a_i}{2} + k_4 t + t^{3/2}P'_1(t) + \frac{z^2}{\sqrt{t}}P'_2(t)$$

$$I(z,t) \;=\; \frac{1}{2}\sqrt{\frac{Dt}{\pi}} \sum_{i=1}^{n-1} a_i + tP_3(t) + \frac{z^2}{\sqrt{t}}P'_2(t)$$

Finally we obtain:

$$F(0,z,t) = 1/2 - \frac{z}{(4\pi Dt)^{1/2}} + \frac{1}{2}\sqrt{\frac{Dt}{\pi}} \sum_{i=1}^{n-1} a_i + tP_3(t) + \frac{z^2}{\sqrt{t}}P'_2(t) + (z/\sqrt{t})^3 k_1$$

to which we could add a term in $exp(-k^2/t)$ accounting for the fact that the expanding region, outside a area of radius k, can no longer be described as the region under the graph of a function. Since this is exponentially decreasing it is much smaller than the three other corrective terms and therefore we will assume they dominate it. It is then easy to calculate the initial speed of the level set $F = 1/2$. We must solve the equation:

$$-\frac{z}{(4\pi Dt)^{1/2}} + \frac{1}{2}\sqrt{\frac{Dt}{\pi}}\sum_{i=1}^{n-1} a_i + t^2 P_3(t) + z^2\sqrt{t}P_4(z,t) + (z/\sqrt{t})^3 k_1 = 0 \quad ,$$

where P_2 and P_4 are bounded by polynomials.
We have:

$$z = (D\sum_{i=1}^{n-1} a_i)t + t^{3/2}P(t) + z^2 P(z,t) + (z)^3/tk_1$$

We then write:

$$z = (D\sum_{i=1}^{n-1} a_i)t + t\epsilon(t)$$

and assume $\epsilon(t)$ is bounded for t small. Then:

$$\epsilon(t) = \sqrt{t}f(t, \epsilon(t))$$

where f is bounded for t small. Knowing that the original F was continuous on $\Re^n * \Re^{+*}$ in (X,z) and t, we immediately deduce from the above that there exist $\epsilon(t)$ satisfying this relation when t is small enough, and such that:

$$\lim_{t\to 0}\epsilon(t) = 0 \quad .$$

This proves[3] that the initial speed of the front is

$$D\sum_{i=1}^{n-1} a_i$$

[3] Since we know that curvature linear-dependent motion is stable, the above is enough to prove that the algorithm converges. The second order perturbation it introduces at each step will thus not have a catastrophic effect after a finite time.

4 Affine Velocity

This algorithm also enable us to perform affine velocity front motion. We must then track at each time step the level $l = 1/2 - \epsilon$, where $\epsilon = v\sqrt{\frac{t}{4\pi D}}$ instead of 1/2.
We will then have:

$$z = vt + (D\sum_{i=1}^{n-1} a_i)t + terms.of.superior.order.$$

therefore the speed of the front will be:

$$v + (D\sum_{i=1}^{n-1} a_i)$$

directed outward. In this case both regions are no longer symmetrical: If we had chosen to diffuse the characteristic function of the other region we would have had to track the level $l = 1/2 + \epsilon$ to obtain the same motion.

Also the level we track is now a function of the time step we use. This procedure is difficult to implement numerically, as opposed to the simplicity of the pure curvature velocity case.

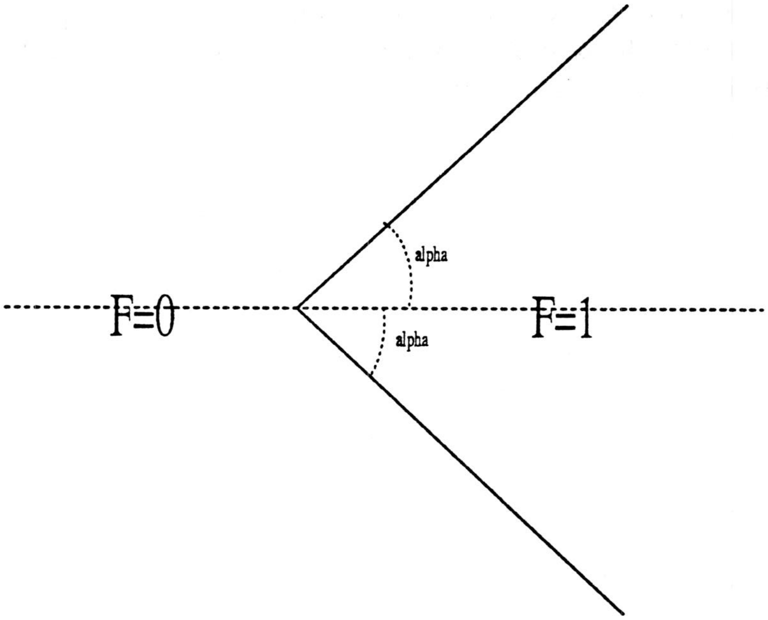
5 Angles and Triple Points

5.1 Angles

One of the advantages of this method is that it handles very easily sharp angles because it doesn't need to calculate the curvature explicitly. The speed of the corner of a sharp angle is obviously infinite, and we have tried to study the second order behavior of those points. In all the following, we will only consider 2 dimensional problems.

5.1.1 Motion of the edge of an angle for a 2 regions problem

We have first tried to calculate an approximation of the solution of the heat operator around a sharp angle, using the explicit solution as we did above. We here use polar coordinates centered at the edge of the angle. The initial expanding area is delineated by the two half-lines $\theta = \alpha$ and $\theta = -\alpha$, i.e. $F(r, \theta, 0) = 1$ for $-\alpha \leq \theta \leq \alpha$, $F(r, \theta, 1) = 0$ otherwise.



Then

$$F(r,\theta,t) = \frac{1}{4\pi Dt}\int_0^\infty \int_{-\alpha}^{\alpha} exp-\frac{(R\cos\phi - r\cos\theta)^2 + (R\sin\phi - r\sin\theta)^2}{4Dt} Rd\phi dR$$

$$F(\sqrt{4Dt}r,\theta,t) = \frac{exp(-r^2)}{\pi}\int_0^\infty exp-R^2 \int_{-\alpha}^{\alpha} exp(2Rr\cos(\phi-\theta))Rd\phi dR$$

This shows that the patterns for all times are homothetic to that of t=1, with ratio $\sqrt{t}$: $F(r,\theta,t) = F(r/\sqrt{t},\theta,1)$. For the motion of the edge of the angle, we calculate F for $\theta = 0$.

$$F(\sqrt{4Dt}r,0,t) = \frac{exp(-r^2)}{\pi}\int_0^\infty exp-R^2 \int_{-\alpha}^{\alpha} exp(2Rr\cos\phi)Rd\phi dR = G(r)$$

And $F(\sqrt{(t)}r,0,t) = F(r,0,1)$. If we denote $r_{1/2}$ the value of r such that $F(r,0,1) = 1/2$, we have:

$$r_{1/2}(t) = \sqrt{t}r_{1/2}$$

Unfortunately the value of $r_{1/2}$ can not be obtained in closed form. It can be numerically computed by solving : $F(r,0,1) = 1/2$.

If we study the evolution of an hyperbola : $y = \sqrt{(\epsilon + px^2)}$, and suppose an hyperbola is a stable by curvature dependent motion (i.e. the front at different times would always remain an hyperbola, with ϵ being a function of t: this is not actually the case) we find a very likely result. We observe the intersection M of this hyperbola with the Y axis. the curvature in M is:

$$\frac{\sqrt{\epsilon(t)}}{p}$$

so its speed should be :

$$Dp/\sqrt{\epsilon(t)}$$

but its ordinate is:$\sqrt{\epsilon}(t)$. We should then have:

$$\partial\epsilon(t)/\partial t = 2Dp \quad ,$$

i.e., $\epsilon(t) = 2Dpt$ and M having a ordinate of

$$\sqrt{2Dpt}$$

which is what we have found above.

5.2 Triple Points, Multiple Points

In case of a triple point we have in fact three regions. We diffuse each one of the three characteristic functions at each time step and then define the three new regions as the set of points where the corresponding function is greater than the two other. In the following we will call that "chopping". Numerical approximations show that, regardless of the initial angles between the tangents to the boundary at the triple point, they very quickly move so as to obtain three angles of 120 degrees. This configuration then remains stable , though the triple point may continue to move or rotate. But this motion is not specific to the triple point and is only induced by the boundaries. (the boundary between 2 regions always tend to straighten out).

5.2.1 Approximation of the Diffusion of an Angle Near the Origin

In order to study the motion of a triple point we will calculate an approximation, for r small, of the diffusion of an angle near the origin, using our results above. If we develop $exp(2Rr\cos{(\phi-\theta)})$into a series for r small, we obtain:

$$F(\sqrt{4Dt}r,\theta,t) = exp{-r^2}(\frac{\alpha}{\pi}+r(\frac{\sin\alpha\cos\theta}{\sqrt{\pi}})+t.s.o)$$

It is even possible to calculate all the terms of this series : (we note $I(\theta,n) = \int_{-\alpha}^{\alpha}(\cos{(\phi-\theta)})^n d\phi$)

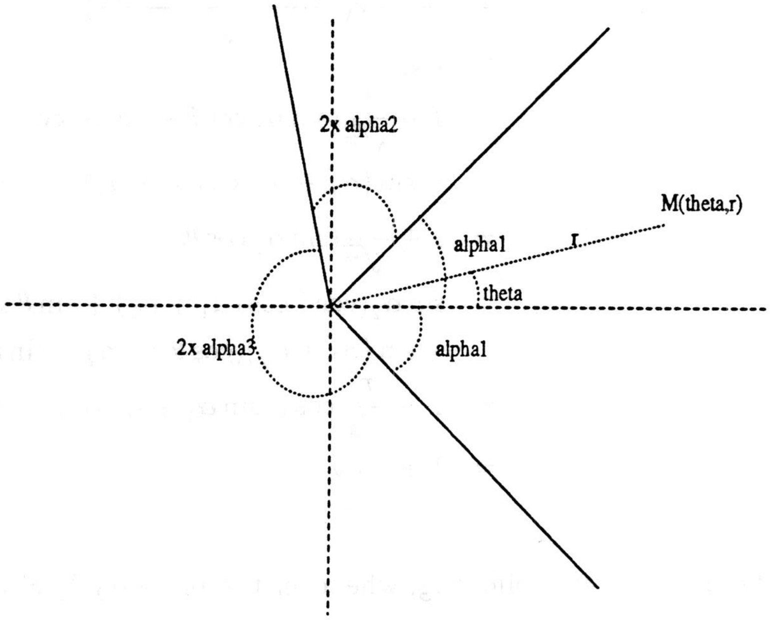
$$F(\sqrt{4Dt}r,\theta,t) = exp{-r^2}(\sum_0^\infty r^{2i}(\frac{i!2^{2i-1}I(\theta,2i)}{\pi(2i)!}+\sum_1^\infty r^{2i-1}(\frac{I(\theta,2i-1)}{2\sqrt{\pi}(i-1)!})$$

which evidently has a convergence radius of $+\infty$.

If we only keep first order terms we obtain:

$$F(\sqrt{4Dt}r,\theta,t) = \frac{\alpha}{\pi}+r(\frac{\sin\alpha\cos\theta}{\sqrt{\pi}})+t.s.o$$

we can now write this for the three different regions around our triple point (we of course approximate the branches with their tangent half-lines)



$$F_1(\sqrt{4Dt}r,\theta,t) = \frac{\alpha_1}{\pi} + r(\frac{\sin\alpha_1\cos\theta}{\sqrt{\pi}}) + t.s.o$$

$$F_2(\sqrt{4Dt}r,\theta,t) = \frac{\alpha_2}{\pi} + r(\frac{\sin\alpha_2\cos(\theta-(\alpha_1+\alpha_2))}{\sqrt{\pi}}) + t.s.o$$

$$F_3(\sqrt{4Dt}r,\theta,t) = \frac{\alpha_3}{\pi} - r(\frac{\sin\alpha_3\cos(\theta-\alpha_2)}{\sqrt{\pi}}) + t.s.o$$

With $alpha_1 + alpha_2 + alpha_3 = \pi$.
We have $F_1 + F_2 + F_3 = 1$ at all times (because the diffusion kernel is linear and $F_1 + F_2 + F_3 = 1$ for $t = 0$). We can easily check that the above approximation keeps this property:

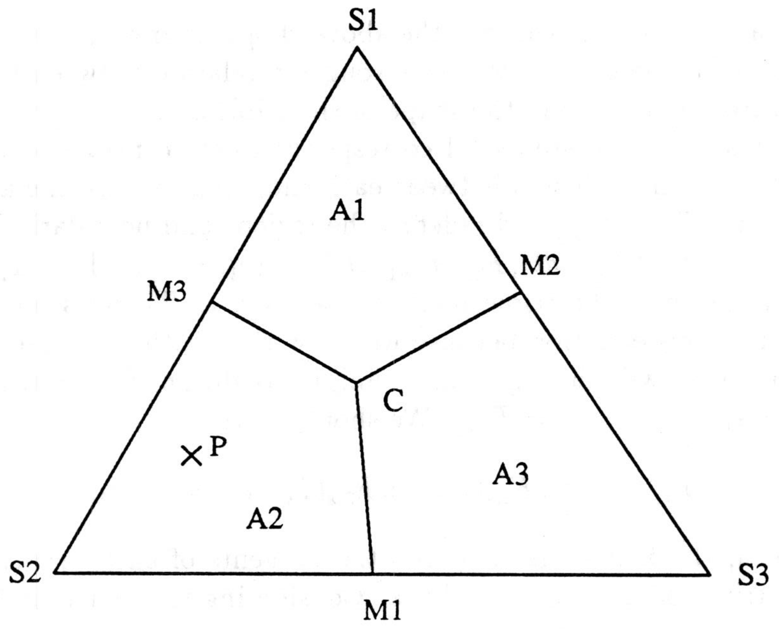
$$\begin{aligned}(F_1+F_2+F_3)(\sqrt{4Dt}r,\theta,t) &= \frac{\alpha_1}{\pi} + r\frac{\sin\alpha\cos\theta}{\sqrt{\pi}} \\ &+ \frac{\alpha_2}{\pi} + r(\frac{\sin\alpha_2\cos(\theta-(\alpha_1+\alpha_2))}{\sqrt{\pi}})\end{aligned}$$

$$
\begin{aligned}
+ &\quad \frac{\alpha_3}{\pi} - r(\frac{\sin\alpha_3 \cos(\theta-\alpha_2)}{\sqrt{\pi}}) \\
+ &\quad t.s.o \\
= &\quad 1 + \frac{r}{\sqrt{\pi}}(\sin\alpha_1 \cos\theta + \sin\alpha_2 \cos(\theta - (\alpha_1+\alpha_2)) \\
&\quad - \sin(\alpha_1+\alpha_2)\cos(\theta-\alpha_2)) + t.s.o \\
= &\quad 1 + \frac{r}{\sqrt{\pi}}(\sin\alpha_1 \cos\theta \\
+ &\quad \sin\alpha_2(\cos\theta\cos(\alpha_1+\alpha_2) + \sin\theta\sin(\alpha_1+\alpha_2)) \\
&\quad - \sin(\alpha_1+\alpha_2)(\cos\theta\cos\alpha_2 + \sin\theta\sin\alpha_2)) + t.s.o \\
= &\quad 1 + \frac{r}{\sqrt{\pi}}\cos\theta(\sin\alpha_1 + \sin(\alpha_2 - (\alpha_1+\alpha_2))) + t.s.o \\
= &\quad 1 + t.s.o
\end{aligned}
$$

This will be useful in the following, whenever the property $\sum F_i = 1$ will be needed.
We will also notice that for $r = 0$, the values of the F_i are exactly equal to $\frac{\alpha_i}{\pi}$ at all times.
Obviously, if we select at each step the greatest of the three functions and redefine the three regions accordingly as we did above, the triple point will be the point of the plane where all three functions are equal, and their value will then be 1/3. Therefore, a necessary condition for obtaining a motionless triple point will be to have three angles of 120 degrees. Otherwise the triple point will move with infinite initial velocity, as described above for a two region sharp angle. We will later try to prove that its motion will tend to bring back all three angles toward 120 degrees.

5.3 Obtaining different shapes of triple points

We can chose other ways of redefining the three regions at each step. A very simple one, which still allows us to obtain any kind of angle distribution, handles $F_1(X,dt), F_2(X,dt), F_3(X,dt)$ as the barycentric coordinates of a point,P , inside a triangle, which we will call the "projection triangle". This triangle is divided in three areas, A1 ,A2 ,A3.



If P(x) is in Ai, then the initial characteristic function for the next iteration will be: $F_i(X) = 1, F_{j_{j \neq i}}(X) = 0$

We have divided the triangle into A1 ,A2 ,A3 by choosing a "centroid point", C, and drawing lines from it to the middle of each side of the triangle. It is very important that those lines reach the middle of each side of the triangle: The sides of the triangle correspond to regions where one of the F_i is infinitely small, i.e. far from the triple point. In those regions the scheme should be equivalent to the above two characteristic function problem, and we then have to select the greatest function at each step. Therefore those lines have to reach the middle of each side. If such was not the case we would obtain affine curvature dependent motion in those areas , and the constant in this affine relation would depend on the the time step we use to diffuse the functions between each chopping. We will also note c_1, c_2 and c_3 the barycentric coordinates of C, the centroid point, in the triangle S1, S2, S3. We suppose that $c_1 + c_2 + c_3 = 1$.

5.3.1 Order Zero Approximation

Computations have shown that the above chopping method can give different shapes of triple points. We have sought a relation between the position of the centroid point C and the shape of the triple point. First we must note that the triple point at step $s+1$ corresponds to the centroid point . If dt is the duration of the diffusion between each chopping (we use the above "projection triangle" to chop, and redefine the regions and boundaries), we have: $(F_{1,s}(T_{s+1},dt),F_{2,s}(T_{s+1},dt),F_{3,s}(T_{s+1},dt)) = (c_1,c_2,c_3)$ where T_{s+1} will be the new position of the triple point at step $s+1$. There is no multiplicative factor in this equation because we know that both $c_1+c_2+c_3=1$ and $\sum F_i = 1$. This will then give us a simple condition if we want the triple point to stay still, i.e. $T_s = T_{s+1}$. We should have:

$$(F_{1,s}(T_s,dt),F_{2,s}(T_s,dt),F_{3,s}(T_s,dt)) = (c_1,c_2,c_3)$$

We name α_i the half-angles between the tangents of each branch emerging from the triple point as drawn above. Considering the fact that for a sharp angle

$$(F_{1,s}(T_s,dt),F_{2,s}(T_s,dt),F_{3,s}(T_s,dt))$$

are independent of dt and are equal to

$$(\alpha_1/\pi,\alpha_2/\pi,\alpha_3/\pi) \quad ,$$

we obtain the following necessary conditions:

$$\begin{aligned}\alpha_1/\pi &= c_1\\ \alpha_2/\pi &= c_2\\ \alpha_3/\pi &= c_3\end{aligned}$$

The experimental validity of these conditions will be discussed below.

5.3.2 First Order Approximations

The above conditions are necessary but at this point are not proven to be sufficient. We have tried, using the above first order approximation, to determine the behavior of the branches. We consider a case where the initial branches are straight half-lines, and use the approximation to calculate the positions of the new branches when the chopping occurs. To this effect we will study the boundary between the first two regions. Points located on this boundary should be projected on the segment joining C and M_3, the middle of [S1-S2]. The barycentric coordinates of M_3 are $(1/2, 1/2, 0)$. The new boundary between 1 and 2 (after the chopping) will then the set of points which are not in area 3 and such that there exists γ:

$$\begin{aligned} F_1 &= \gamma/2 + (1-\gamma)c_1 \\ F_2 &= \gamma/2 + (1-\gamma)c_2 \\ F_3 &= (1-\gamma)c_3 \end{aligned}$$

As $\sum F_i = 1$, the 2 first imply the third equation, this being also true if we replace the F_i by their first order approximations. Using the homothetic properties of a diffusing angle we will note: $R = \sqrt{4Dtr}$. the first order approximation then gives:

$$\begin{aligned} F_1(R,\theta,t) = (1-\gamma)/2 + \gamma c_1 &= \frac{\alpha_1}{\pi} + r(\frac{\sin\alpha_1 \cos\theta}{\sqrt{\pi}}) + t.s.o \\ F_2(R,\theta,t) = (1-\gamma)/2 + \gamma c_2 &= \frac{\alpha_2}{\pi} + r(\frac{\sin\alpha_2 \cos(\theta - (\alpha_1 + \alpha_2))}{\sqrt{\pi}}) + t.s.o \\ F_3(R,\theta,t) = \gamma c_3 &= \frac{\alpha_3}{\pi} - r(\frac{\sin\alpha_3 \cos(\theta - \alpha_2)}{\sqrt{\pi}}) + t.s.o \end{aligned}$$

These equations are linked because $\sum F = 1$ so we will only consider the first and third lines. We obtain:

$$\gamma = \frac{\frac{\alpha_3}{\pi} - r(\frac{\sin\alpha_3 \cos(\theta-\alpha_2)}{\sqrt{\pi}})}{c_3}$$

and

$$\frac{\frac{\alpha_3}{\pi} - r(\frac{\sin\alpha_3 \cos(\theta-\alpha_2)}{\sqrt{\pi}})}{c_3}(c_1 - 1/2) + 1/2 = \frac{\alpha_1}{\pi} + r(\frac{\sin\alpha_1 \cos\theta}{\sqrt{\pi}})$$

This equation (coupled with the two others we would obtain writing this for the two other branches) has obviously no solution for r small if the order zero conditions cited above are not satisfied. If such was the case, the triple point would move with infinite initial velocity, and any computation of its motion would heavily depend on the time step used for the first iteration. Therefore we will assume that those conditions are fulfilled, and consequently:

$$\alpha_1/\pi = c_1, \alpha_2/\pi = c_2, \alpha_3/\pi = c_3$$

We then have:

$$(1 - \frac{r}{c_3}(\frac{\sin\alpha_3\cos(\theta-\alpha_2)}{\sqrt{\pi}}))(c_1 - 1/2) + 1/2 = c_1 + r(\frac{\sin\alpha_1\cos\theta}{\sqrt{\pi}})$$

Finally r disappears and we have:

$$\frac{(c_1 - 1/2)}{c_3}(\sin\alpha_3\cos(\theta-\alpha_2)) = \sin\alpha_1\cos\theta$$

We can then obtain $\tan\theta$

$$\frac{(c_1 - 1/2)}{c_3}\sin\alpha_3(\cos\theta\cos\alpha_2 + \sin\theta\sin\alpha_2) = \sin\alpha_1\cos\theta$$

$$\cos\theta(\sin\alpha_1 - \frac{(c_1 - 1/2)}{c_3}\sin\alpha_3\cos\alpha_2) = \frac{(c_1 - 1/2)}{c_3}\sin\alpha_3\sin\alpha_2\sin\theta$$

$$\tan\theta = \frac{(\frac{c_3}{(c_1 - 1/2)}\sin\alpha_1 - \sin\alpha_3\cos\alpha_2)}{\sin\alpha_3\sin\alpha_2}$$

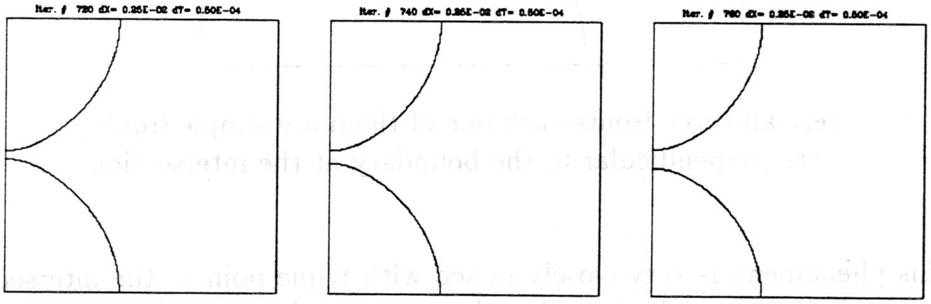
This differs from $\tan\alpha_1$, except in some cases(one of them being $c_1 = c_2 = c_3 = 1/3$). It thus doesn't prove that there can be a still triple point but rather induces us to think the contrary. We though must remember that our approximation was obtain by developing $F(\sqrt{4Dt}r, \theta, t)$. If we consider $F(R, \theta, t) = F(\sqrt{4Dt}r, \theta, t)$, the above will only be accurate for r small, or $R/\sqrt{(4Dt)}$ small. What we have done does not apply if the successive positions of the triple point are at a distance of $\sqrt{(4Dt)}$, or more.
We will now compare these results with those obtained by computation.

6 Computational Implementation of the Algorithm

This algorithm has been programmed. A five point Laplacian was used to compute diffusion, on a grid of 300X300 points (this size was chosen in order to obtain a good precision, while keeping the program reasonably fast. We also used a 200X200 grid) . The boundary conditions, which as we will see are very important, were of null flux (homogeneous Neumann conditions). We have tried several different values for the time step used in the diffusion kernel, the number of iterations between each chopping, as well as the position of the centroid point. The two first affect the stability and the accuracy of the program, while the third enabled us to test our calculations concerning triple points.

6.1 simple curves

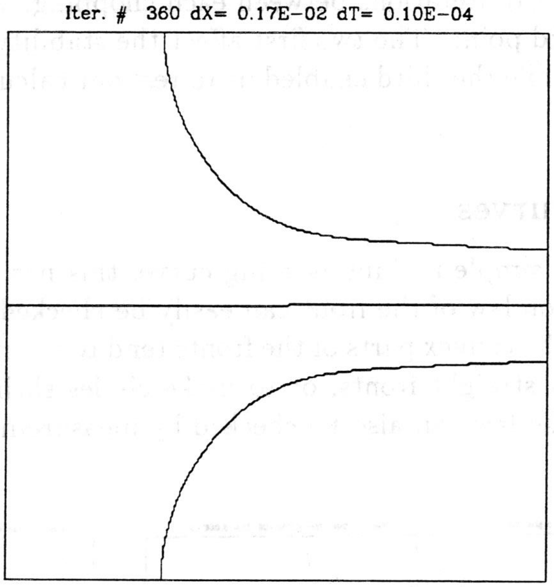
In the case of a simple non-intersecting curve, this has yielded very good results. The motion law of the front can easily be checked to be a curvature dependent one. The convex parts of the fronts tend to shrink, and the general effect is to obtain straight fronts, or to make circles shrink and disappear. The linearity of the law can also be checked by measurement.



here we can see two collapsing circles.
On more complex fronts we can check the correlation
between curvature and velocity. See the end of the report.

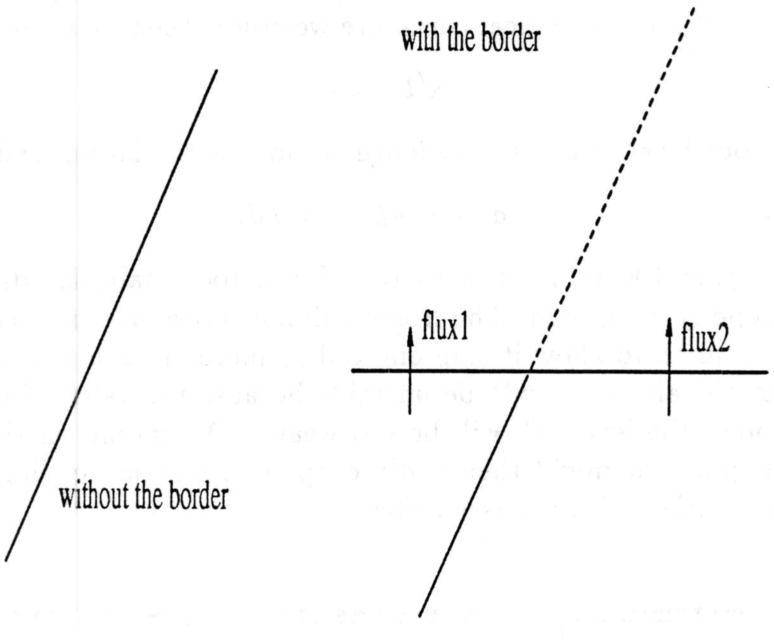
6.1.1 Boundary problems

However, an important problem occurred. While the program gives excellent results as long as the curve remains confined inside the grid, it has the greatest difficulties to handle curves of infinite extension or whenever it intersects with the boundary of the grid. The only equilibrium available for such an intersection was found to be a 90 degree angle. One iteration is enough to come to this equilibrium, because the velocity of the front is infinite for all angles other than 90 degrees.



here all three fronts(each one of them are simple fronts)
are perpendicular to the boundary at the intersection

This phenomena is very closely linked with triple points. the intersection between the front an the boundary has to be understood as a triple point. Ideally we would like that those intersection to be stable in all positions. This would avoid any interference between the boundary and the inside of the grid. To understand this, we will first consider a straight front, and then consider the effect of adding a boundary to this front.



Adding a boundary will make both flux1 and flux2 null, and introduce a asymmetry: In this case flux1 was negative and flux 2 was positive. The smaller angle(1) will then tend to become bigger, as the area where $F_1 > 1/2$ will decrease in surface, and the reverse will happen to area 2. This will only stop when both angles reach 90 degrees.

Very little can be done to solve that problem. One solution would be to introduce a flux through the boundary that would be calculated to overcome this problem. We would then have to compute the direction of the branches at each intersection with the boundary and then introduce the corresponding flux . The flux would be equal to flux 1 and flux 2 of the above drawing. However the algorithm would then be much more complex, and it would then lose the advantage it has other front tracking methods. Therefore we tried to diminish the impact of this phenomena by increasing the size of the grid, and changing the initial data(see below).

6.1.2 Stability

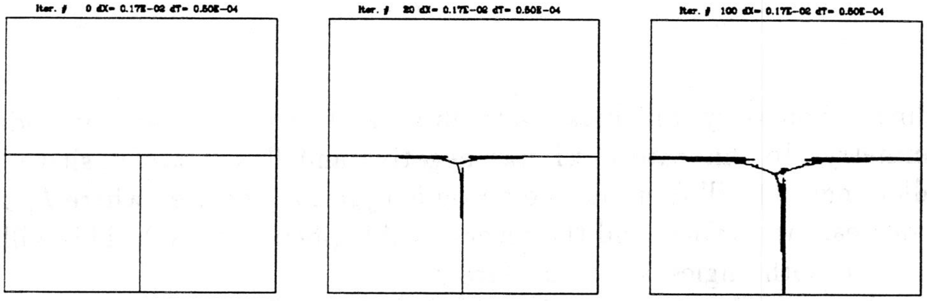
The parameters of the program have to be adjusted in order to make it stable and efficient. Mainly, the time τ step used for diffusion between each chopping, and the size of the elementary cells of the spatial grid have to be compatible. If ρ is the typical curvature we study, then we must have:

$$\sqrt{D\tau} << \rho$$

otherwise our local analysis is no longer valid(τ is too large), and

$$dx << \rho D\tau << kdx$$

with k ranging 100 to 1000 or more. If τ is too small, the diffusion will not be properly computed. The fronts will not move, because the time step will be too small to allow it skip one cell or more at each rechopping. If τ is too big, the algorithm will be unstable because the value of the slope of $F(X)$ around the level 1/2 will be too weak . Depending on the accuracy of the computer, it might then badly chop the function, leading to uneven boundaries, "islands" etc... (see below)


Iter. # 0 dX= 0.17E-02 dT= 0.50E-04
Iter. # 20 dX= 0.17E-02 dT= 0.50E-04
Iter. # 100 dX= 0.17E-02 dT= 0.50E-04

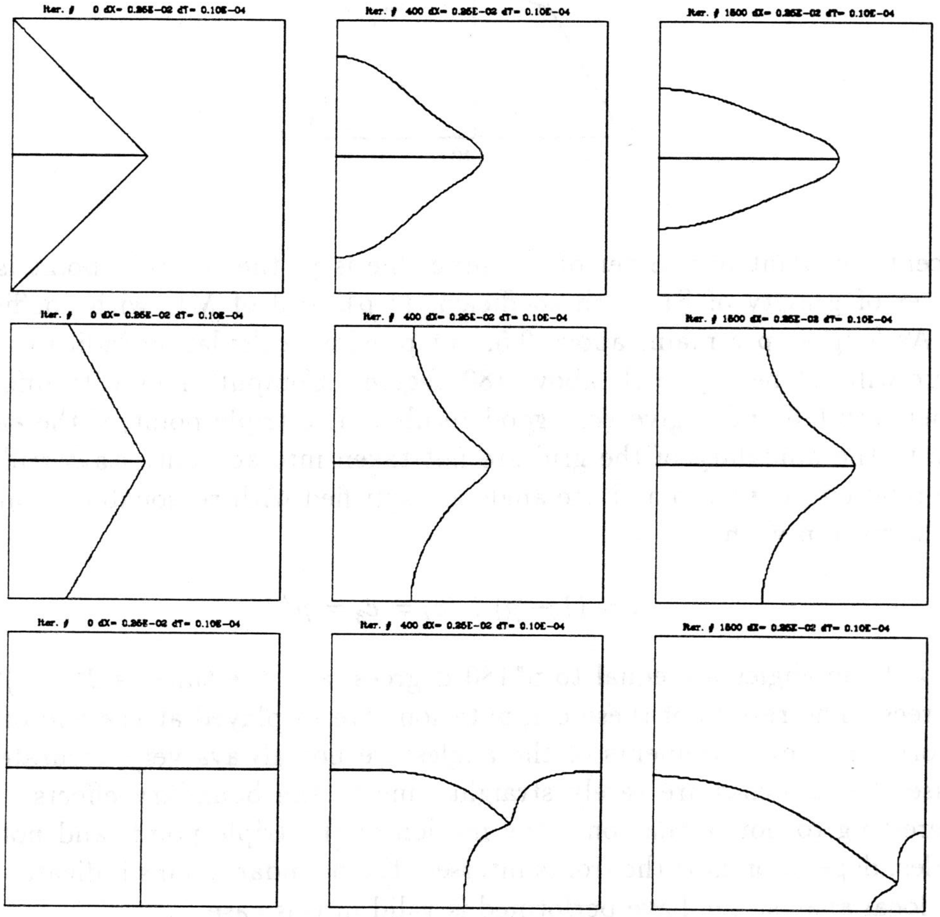
unstable choice for τ: $\tau = 0.00005$, grid of 300 points

The program we use to compute diffusion itself using iterations, using a much smaller time step than the "chopping" iterations. It may also be unstable , if this second time step is to big.

6.2 Experimental Results for Triple Points

6.2.1 Symmetrical Chopping

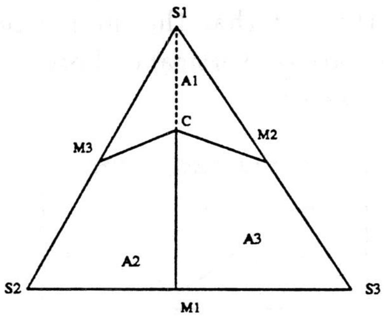
We have first run the program with the centroid point being the center of gravity of the triangle. This is equivalent to select at each chopping the greatest function. This leads to a 3 x 120 degrees triple point. Here the correlation between our above calculations and the simulation is perfect. The triple point moves across the grid. Though this has not been proved, this motion is clearly the consequence of "tension" occurring at the boundary of the grid, as exposed above. Hence, the motion of the triple point, an the shape of the branches outside the immediate vicinity of the triple point can be greatly affected by the fact that the intersection between the front and the boundary of the grid are on vertical or horizontal parts of the boundary, as the following pictures show:



All three cases have been computed with the same chopping method. (The chopping was not symmetrical, and the centroid parameter, defined on the following page, was p=0.32 in all three cases.)

6.2.2 Triple Points with Angles Below 180 Degrees

In order to keep the program quick and simple not all kinds of centroid points have been computed. It has only been allowed to move on the segment linking one of the vertices of the triangle to the the middle of the opposite side:



If centroid point parameter of the input-file is p, the centroid point is the center of gravity of S1, with coefficient (1-p), and of M1, with coefficient p. As long as p remains above 0.5, our previous calculation indicates that there will not be any angle above 180 degrees. Computations with different p between 0.5 and 1 give very good results. The triple point, if the effects due to the boundary of the grid are not taken into account, stays still and the relation between p and the angle are satisfied with reasonable accuracy. For a given p we have:

$$c_1 = (1-p)\ ,\quad c_2 = c_3 = p/2$$

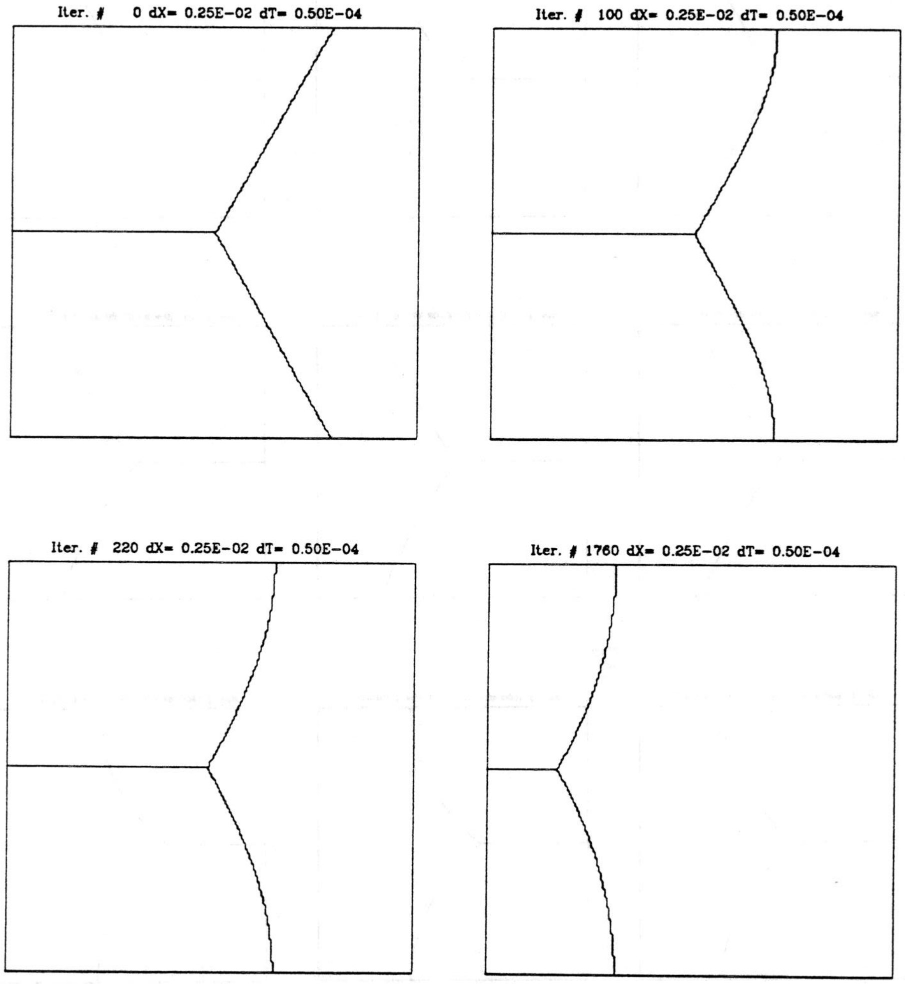
Two of the angles are equal to p*180 degrees, and the third is 2*(1-p)*180 degrees. The results of these computations are displayed at the end of this report. The measurements of the angles are not always very accurate because the branches are rarely straight, due to the boundary effects. It is interesting to notice that only the motion of the triple point, and not the angles, depend on how the fronts intersect the boundary. This indicates that the local analysis we have performed is valid in this case.

6.2.3 Triple Points with Angles Above 180 Degrees

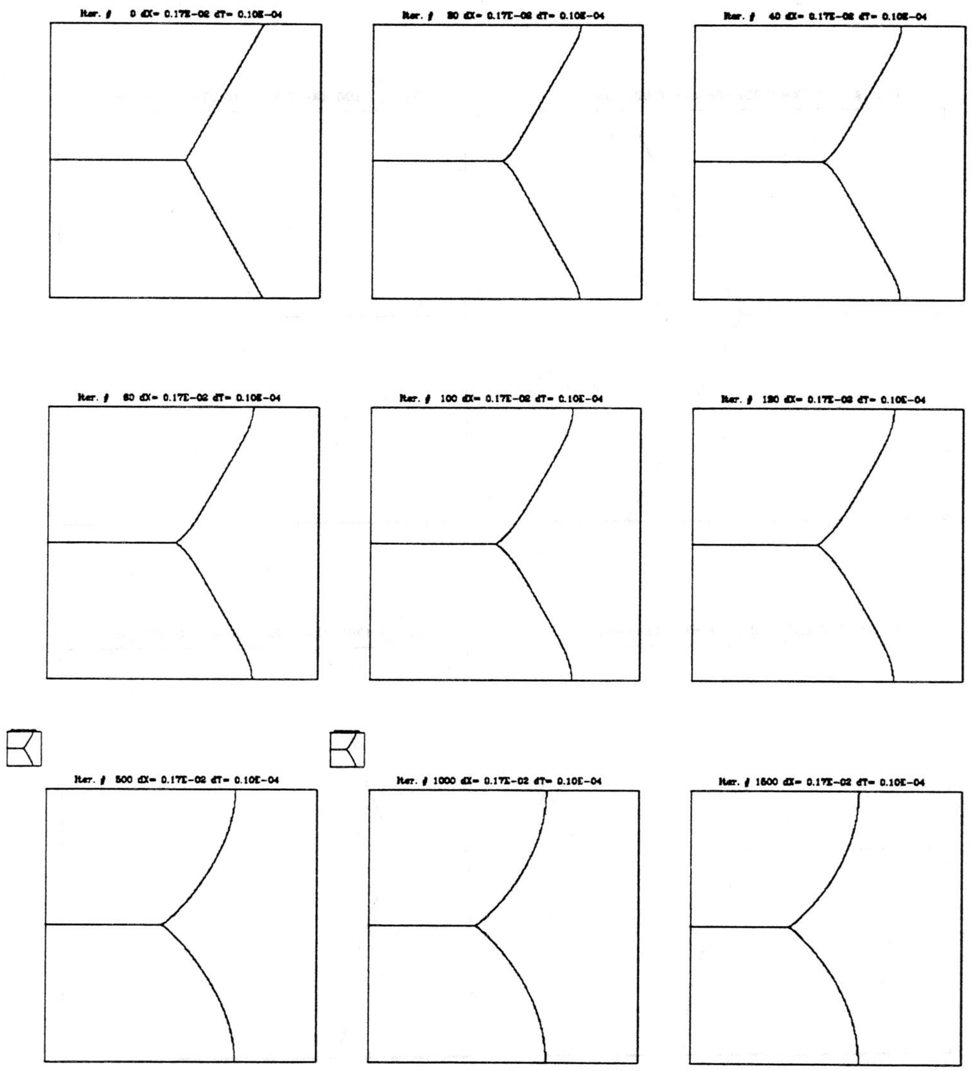
When the parameter p is in [0,0.5], the simulations do not yield a still triple point. When p is under 0.40, the triple point moves rapidly, and its velocity increases as p decreases to 0. The shape of the triple point itself is different. It becomes round, and the largest angle does not seem to go over 180 degrees. It is difficult to measure it, because the branches of the front are not straight, and two of them have a very high curvature near the triple point. The round shape of the triple point is clearly related to its motion. It is has not been completely determined whether this kind of triple point can reach an equilibrium, and stop its motion. The grid we used had to stay of reasonable size, and, most important, the boundary problems could have prevented it from reaching an equilibrium. It is also possible that this method does not converge when the time step tends to 0, or, in other words, that the behavior of the triple point depends upon the time step, as in the affine velocity motion we described above. We have tried the same initial cases with different time steps. It has given slightly different results, without really confirming this last statement. The results of the computation are displayed in the following pages.

RESULTS OF THE COMPUTATION

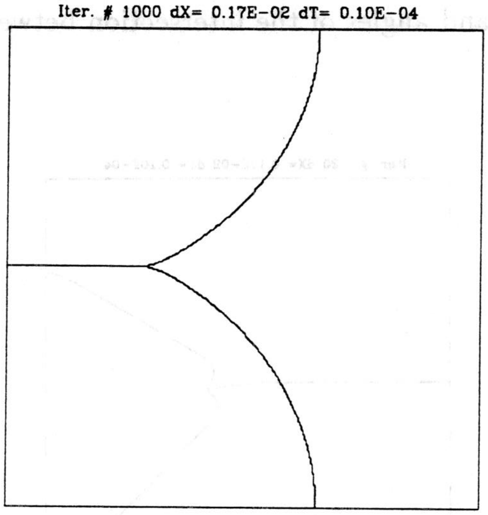
We here see the result of a computation with p=0.66, which should and does give three angles of 120 degrees. The fronts are first change shape to give 120 degrees angles and 90 degrees intersections with the boundary, then the shape of the triple point doesn't evolve any more and it is only translated due to boundary effects.

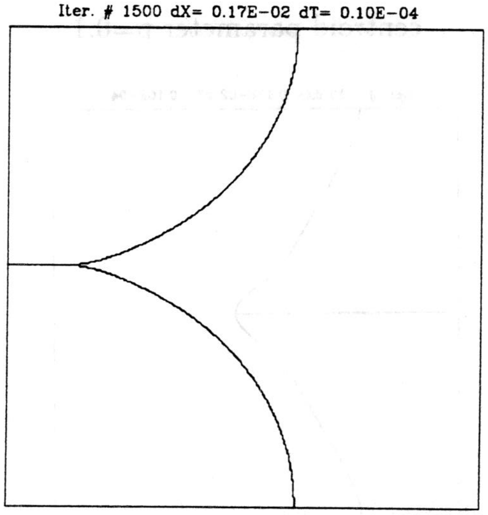


The following shows computations with initial branches of +45 and -45 degrees. The centroid point parameter is 0.80, which should give a theoretical (1-0.8)*360=72 degrees angle. The final angle is approximately 85 degrees. The triple point is then almost still.

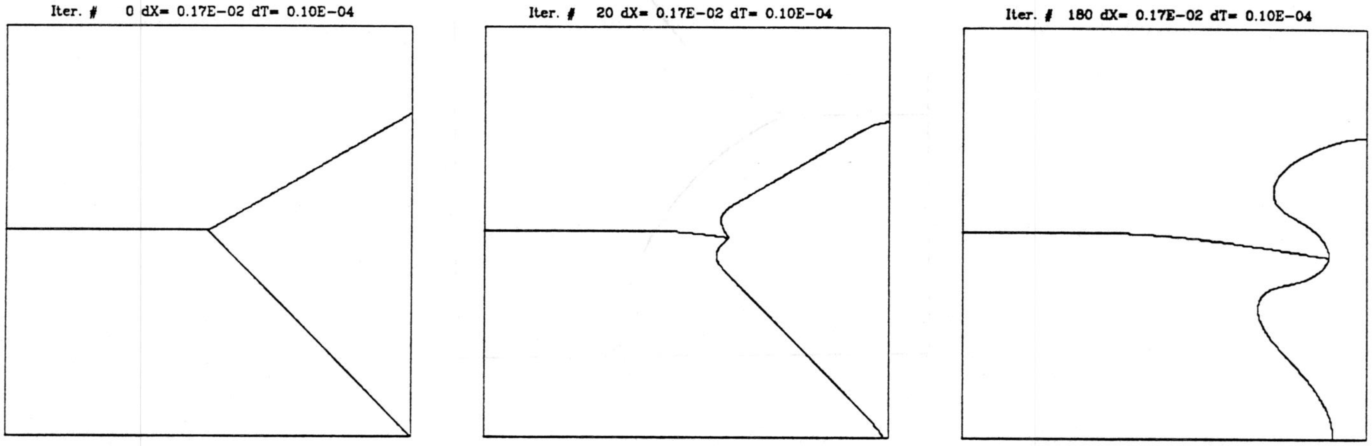


We here see the final results of computations with p= 0.90 and 0.95. They theoretically give angles of 36 and 18 degrees. The actual angles are 38 and 18 degrees.

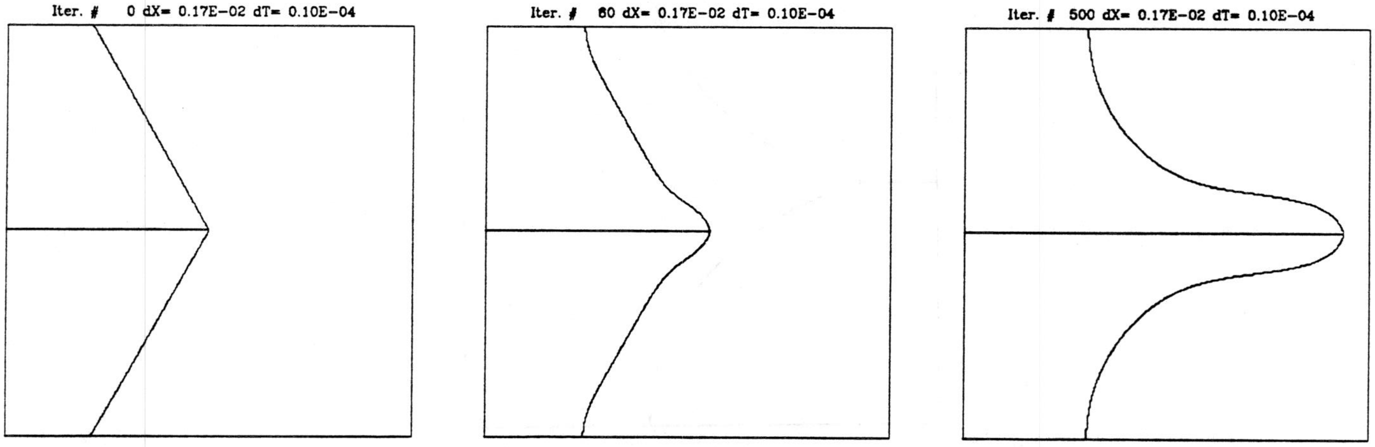




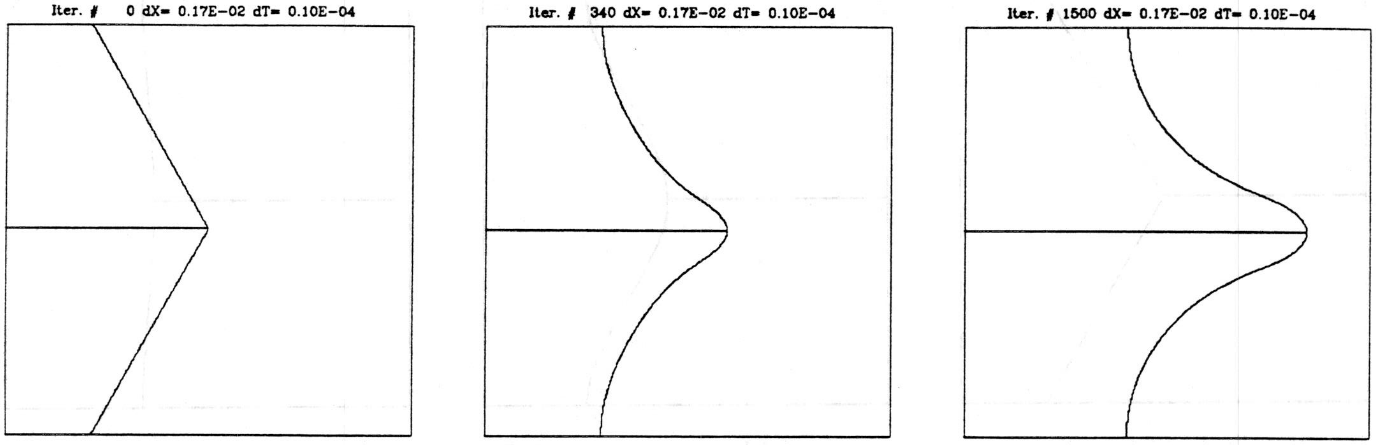
When p is between 0 and 0.5, the triple point starts having a round shape, and it becomes impossible to measure the angles. In fact that round shape could be interpreted as two 90 degrees angles and one 180 degree angle. however the shape of the triple points is still affected by the centroid parameter, as can be observed in the following. This round shape is not an effect of the boundary problems: it occurs almost instantly and is present regardless of the position and angles of the intersection between the front and the boundary.



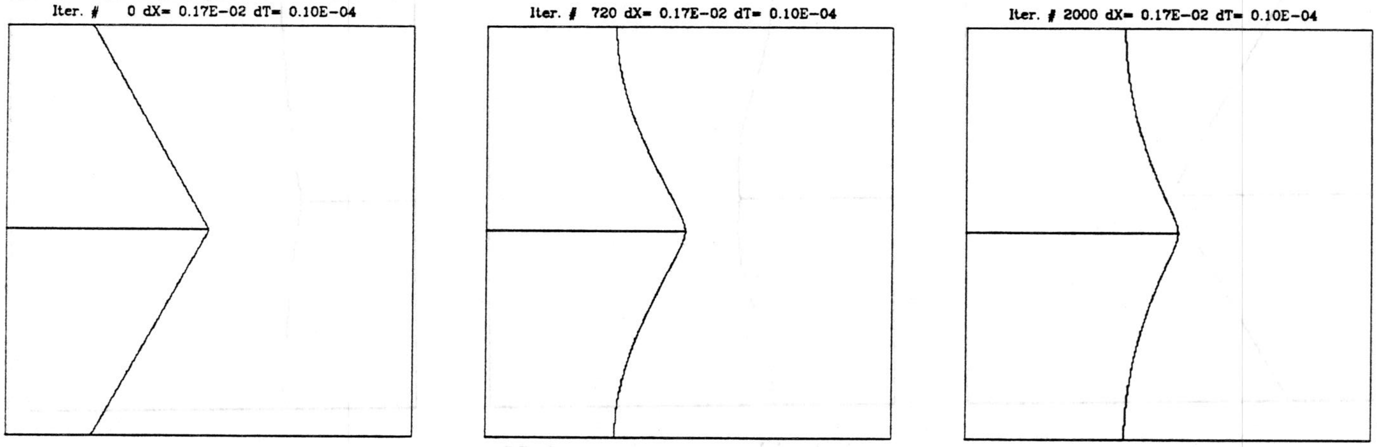
centroid parameter p=0.1



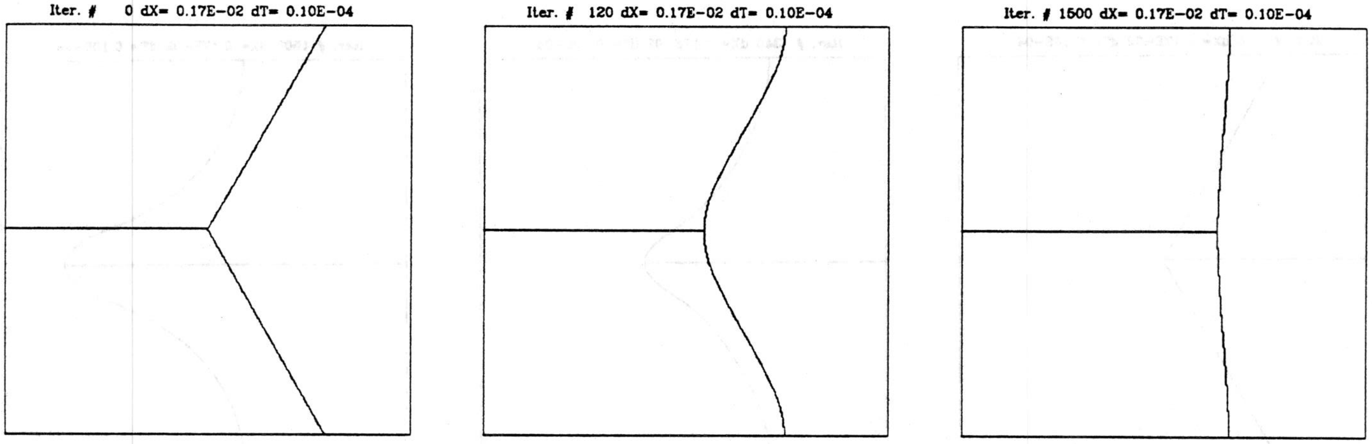
centroid parameter p=0.2



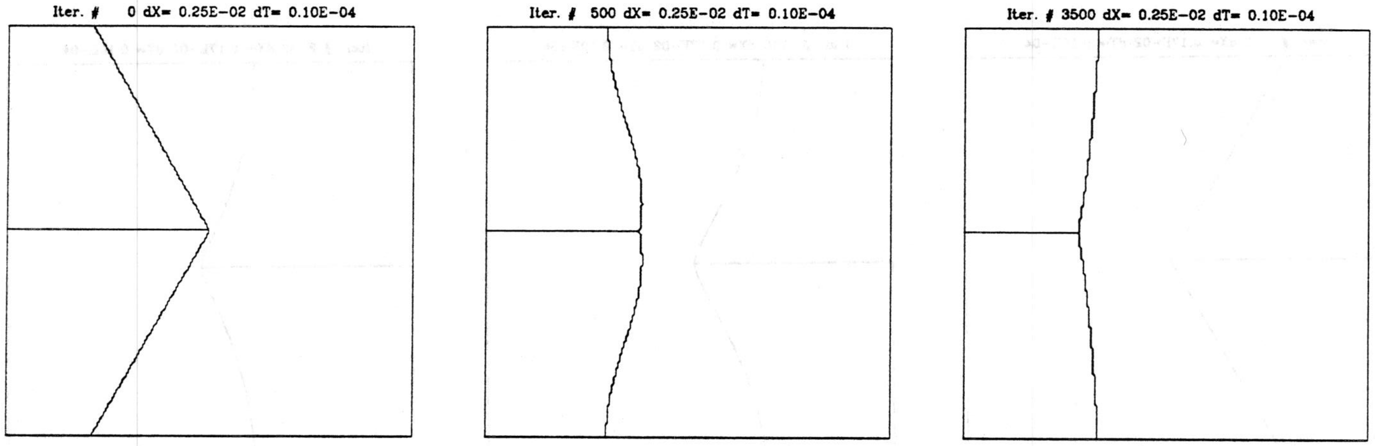
centroid parameter p=0.3



centroid parameter p=0.4



centroid parameter p=0.5



centroid parameter p=0.55

for p=0.5 we have a well defined 180 degrees angle. For greater values of p we have well defined angles. There is a change in the behavior of the triple point for p=1/2.