Clarification of the Relative Minimum Distance (RMD) Theorems and the real-time arclength algorithm

Valere Huypens

Clarification of the Relative Minimum Distance (RMD) Theorems and the real-time arclength algorithm, DOI: 10.13140/RG.2.2.36835.84009

Valere Huypens, Member IEEE

Aen den Hoorn, App. 8 bus 1, 3440 Zoutleeuw, Belgium

email: huypens@telenet.be

Abstract

The well-known Bresenham and the midpoint algorithm of Pitteway and Van Aken belong to the RMD algorithms. The RMD theorem proves why they have the highest accuracy and the fastest real-time execution time. Nowadays the RMD's have constant feedrate and they even measure the rounded arclength in real time. That was until now a main property of the Pythagorean-hodograph curves of Farouki, which also have applications in offsetting curves. The RMD algorithms implement the basic offset curves from the connected points, the implementation with the high degree PH-curves is complex. This paper proves that everyone can generate a "classic" OoA error and that everyone can correct that OoA error at an amazingly straightforward way, and furthermore the OoARange is inversely proportional with the rounded radius. Therefore, rejecting RMD algorithms for the sake of OoA is hair-splitting.

Highlights

- The application of the new RMD-concept: the calculation of the *distance* to a line is much simpler and faster than the calculation of the *distance* to a curve.

$$\overbrace{\text{Sign}[\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v]]}^{\textbf{curve}} \equiv \text{Sign}[(1-\varepsilon_E)*\mu_E*\tau_E]\overbrace{*\text{Sign}[(\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v])]}^{\text{polar } \textbf{line} \text{ of } \frac{\mathbf{P}_u+\mathbf{P}_v}{2}}$$

- The RMD algorithms use the midpoint or the twopoint decision functions
- The RMD algorithms apply to quadratic equations and higher degree implicit algebraic equations
- The RMD algorithms have constant feedrate and measure the arclength of the curves
- The Offset Curves use the same RMD algorithms
- The feedrate, accuracy, flexibility, and execution speed of the RMD algorithms are much better and smoother than those of the sampled-data algorithms
- The real-time implementations of the RMD algorithm are much simpler than the real-time implementations of the Pythagorean-hodograph (PH) curves
- The RMD's can be converted to real-time constant feedrate listSIM-algorithms which can be integrated into rigid simplified CNC machine tools

Keywords

Bresenham's algorithms • Distance • Offsets • Arclength • Constant feedrate CNC-interpolation • Algebraic curves • Trident curves • Reference Pulse IPOs

Mathematical Style

The index of new equations is (Eq integer) and the reference to these equations is (integer). The references to equations of other papers # are [#; (number)] or [#; (number ; number)] or [#; number–number] or [#; § section index], but the references to this paper use [...] without #. The notations $\boldsymbol{\rho}_B^2$ means $\boldsymbol{\rho}_B \bullet \boldsymbol{\rho}_B$; $\triangleq$ means "by definition"; Δ is the grid distance; in practice Δ equals one.

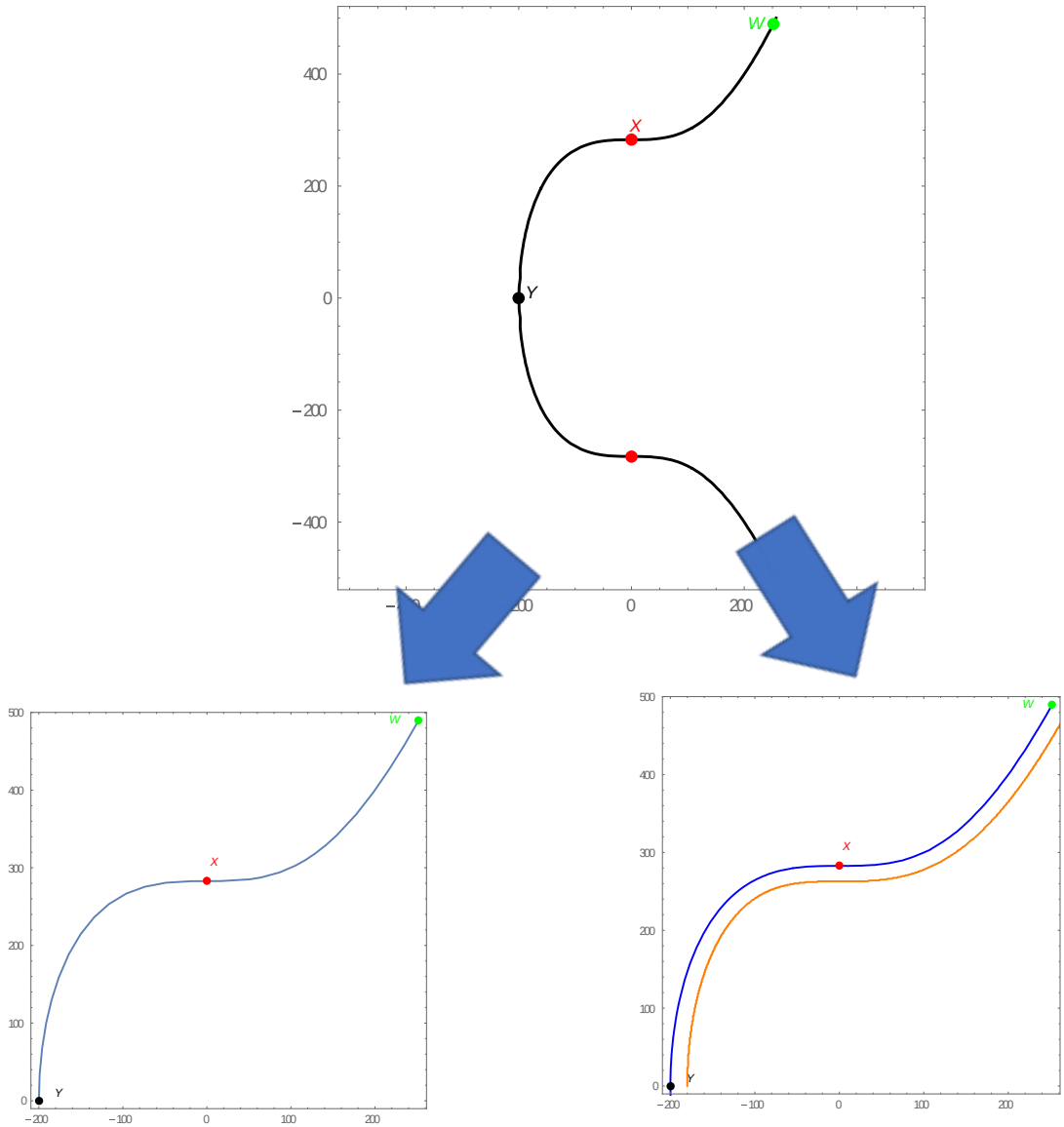
All quadratic and algebraic equations are implicit equations. The equations are written and checked with "Mathematica."

RegionDistance is a time-consuming function measuring the distance to a general QSIC defined in ImplicitRegion. To obtain the meaning of an unknown function, google: "Mathematica and <the name of the unknown function." For the scientific style, this paper uses the Standard Form instead of the Traditional Form of Mathematica, therefore the reader can reproduce the algorithms and check the results.

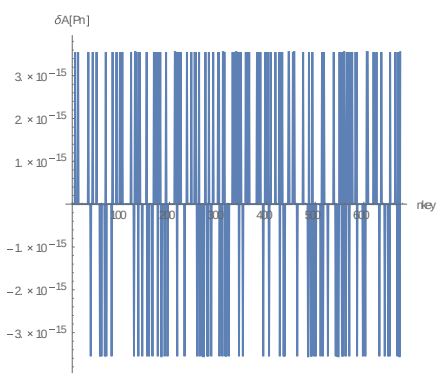
The values in all tables are approximated with N[expr,001].The source code of the programs is in .cdf , .nb or .pdf format. You can open the programs in .cdf format with "Wolfram CDF Player," but you cannot change them. The programs in .nb format need "Mathematica.

Clarification of the Relative Minimum Distance (RMD) Theorems and the real-time arclength algorithm, DOI: 10.13140/RG.2.2.36835.84009

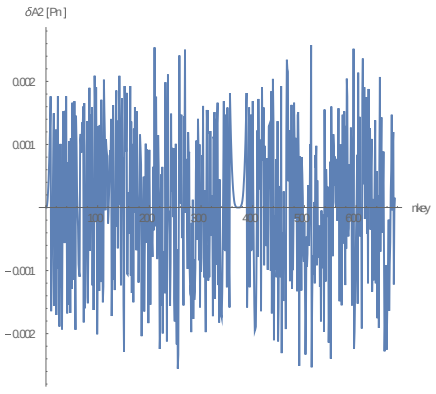
Graphical abstract of the Mordell Curve ≡ {Y, X, W} and its offset
Mordell $F[x,y] = x^3 - SC * y^2 + 8 * SC^3$ with $\Delta = 1$, $SC = 100$, $\delta Offset = 0.2 * SC$



Mordell curve, twopoint curve, and twopoint Offset curve



Offset Errors δA



Offset Error δA2

Abbreviation

Alaska	programming language Alaska Xbase++ version 1.9 from Alaska Software, Germany
BRES-IPO	the collective noun of the midpoint- and twopoint method-IPO
Candidate points	candidate points { $\mathbf{P}_7$, $\mathbf{P}_6$, $\mathbf{P}_4$, $\mathbf{P}_2$, $\mathbf{P}_5$, $\mathbf{P}_3$, $\mathbf{P}_1$ } (Fig. 1) or { $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_1$ } (Fig. 2), ordered along decreasing priority
8-Connected	each best discrete point in a square grid, with given direction has 3 candidate points, and 8 candidate points without the given direction
26-Connected	each best discrete point in a cubic grid, with given direction has 7 candidate points, and 26 candidate points without the given direction
Cost	the sum of the squared errors of the discrete points for a curve
dErr[P_n]	absolute perpendicular distance of the point P_n to a curve, also called the error; measured with the time-consuming function RegionDistance
Extreme point	tangential point $\mathbf{P}_{exti}$ with tangent parallel to the main axis
General Conic	$F_p = FP[\mathbf{P}] = \mathbf{P} \bullet \mathbf{G}[\mathbf{P}] + W[\mathbf{P}] = F[x, y] = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} A_1 & D_1 & I_1 \\ D_1 & B_1 & J_1 \\ I_1 & J_1 & M_1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ (1A)
IPO	incremental step interpolation algorithm
LSD	least square distance or LSD algorithm
Mathematica	the technical computing system of Wolfram Research Illinois
MaxErr	the maximum error of the discrete connected points for a curve
midpoint-IPO	IPO based on the midpoint RMD method
Monotonic dir.	The monotonic direction $\mathbf{T}_{md} \triangleq \mathbf{P}_{ext2} - \mathbf{P}_{ext1}$ and $\{Sx, Sy\} \triangleq Sign[\mathbf{T}_{md}]$.
nkeyM	the number of connected points of the curve
npuls	real-time calculated arclength of an 8- or 26-connected curve also the number of while $AC_{cs} > 0$ loops of the PRM-cs algorithms in the
NURBS	non-uniform rational B-spline
OoC	out-of-control [5, Chapter 6], the tangential direction is not compatible the monotonic direction
OoA	out-of-accuracy [5, Chapter 7], the selected move is wrong
Pole	the point $\mathbf{P}_m$ defining the coefficients of the polar line
Polar line	each conic or quadric has a polar line with respect to an arbitrary point $\mathbf{P}_m$
PRM	pulse-rate-multiplier
PRM-cs	converts the connected LSD- and RMD-IPOs to constant feedrate IPOs
PRM-IPO	major axis IPO without division or multiplication
QSIC	intersection curve of two quadrics
Ref-P-IPOs	Reference Pulse IPOs: if date ≥ 2015 real-time RMD-IPOs else BRES-IPOs
RMD	RMD algorithm based on the **R**elative **M**inimum **D**istance of two candidate points $\{\mathbf{P}_u, \mathbf{P}_v\}$ to the curve, it always generates an 8- or 26-connected curve
Midpoint RMD	RMD using the **P**olar **L**ine of the conic, quadric, or QSIC with respect to the midpoint of the candidate Points → midpoint method
Twopoint RMD	RMD using the difference $(FP[\mathbf{P}_u])^2 - (FP[\mathbf{P}_v])^2$ of the squared residues of the implicit algebraic polynomial FD[**P**] → twopoint method
Singular	an extreme point without any direction
Twopoint-IPO	IPO based on the twopoint RMD
VIRT-IPO	IPO based on selecting the candidate point with the LSD with integrated priority

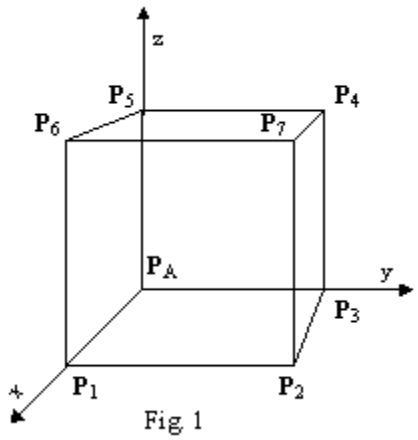


Fig. 1

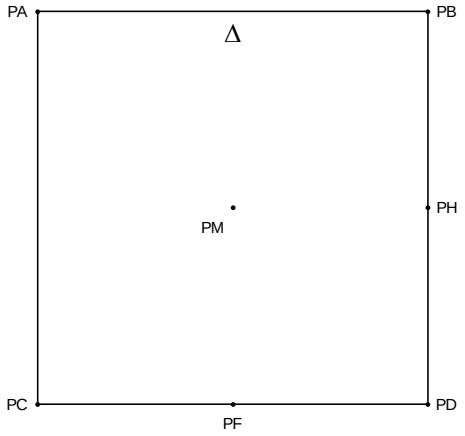


Fig. 2. Candidate points of an 8-connected curve

1. Introduction

The RMD algorithms explain the midpoint and twopoint algorithms.
Essential the RMD algorithms calculate the "best" 8- or 26-connected points of the 2D- or 3D-curve, using the sign of the decision functions of two candidate points. Starting from a best point and a given monotonic direction, the number of possible best points or candidate points is three or seven for an 8- or 26-connected curve. The RMD algorithm does not calculate the minimum distance of all the candidate points, it compares pairwise [3; 4, §2.5] the decision function and therefore, the selected best point is the candidate point with the minimum distance to the curve, without calculating the true distance to the curve. The decision function measures the relative squared distance of two candidate point to the curve using the relative squared distance of these candidate point to the polar **line** of the quadric (or the quadratic approximation in the range of these candidate points) with respect to the midpoint of these candidate points [5, §5]. This explains the accuracy and the fast calculation of the RMD algorithms . The Relative Minimum Distance Theorems are valid for 2D quadratic and algebraic curves but also for 3D quadratic and algebraic curves [1; 2; 3; 4].
Cascading an ultra-fast 3-lines algorithm "PRM-cs" to the 8-connected midpoint or twopoint algorithms quickly converts these algorithms to real-time constant feedrate algorithms and [1; 2; 3; 4] extend this to 3D quadratic and algebraic curves. Paper [4] slightly corrects the PRM-cs for 2D-conics and 3D-curves.
Since the publication of Bresenham's circle algorithm in 1965 [10], a plethora of papers discussed this twopoint algorithm and the midpoint algorithm of Pitteway [13]. Counter examples showed that the twopoint algorithm made an error not made by the midpoint algorithm and counter examples showed that the midpoint algorithm made an error not made by the twopoint algorithm. This lasted until the publication, in 1985 of Van Aken [14], which said that the error of the midpoint method was smaller than $0.5\,\Delta^2$ and that experiments showed, without rigorous proof, that the midpoint algorithm had less errors. This was general accepted, and the discussions disappeared. The publication [5] in 2015 rigorous proved that both methods are the same for conics and that both algorithms make OoA errors but that the midpoint algorithm has less OoA errors than the twopoint algorithm. Both algorithms are RMD-algorithms (the reference pulse IPOs before and after 2015 are different), and it seems that these simple IPOs were extremely difficult to comprehend. The 2D-extension to general conics-IPOs [5], the 3D-extension to QSIC-IPOs [3; 4], the extension to constant feedrate IPOs [1; 2; 3; 4] and the extension to offset-curves [4] make the comprehension of these IPOs even more difficult.
According to Rida T. Farouki [17] "It is impossible to parameterize any real plane curve, other than a straight line, by rational functions of its arc length". I did not realize that the extension to constant feedrate curves [1; 2; 3] and to offset curves [4] was as important, it

means that the result of the real-time constant feedrate RMD algorithm which generate a discrete connected curve $\mathbf{r}[\,T_c\,]$ with feedrate $F_c \equiv \frac{1}{T_c}$, connected point $\mathbf{P}[i*T_c\,]$ and gradient $\mathbf{G}P[i*T_c\,]$, can be written as a discrete connected curve currently dependent on the rounded arc length nputs$[\,T_c\,]$ or

$\mathbf{r} = \text{listSIM} = \{\text{nputs}[i*T_c\,], \mathbf{P}[i*T_c\,]\}\big|_{i=1}^{i=nkeyM}$ [1; 2, §7.2, §7.3] and for offsets as

$$\mathbf{r}_{offset} = \text{listSIM}_{offset} = \{\text{nputs}[i*T_c\,], \mathbf{P}[i*T_c\,], \mathbf{G}P[i*T_c\,]\}\big|_{i=1}^{i=nkeyM} \quad (55)$$

A 2D- and 3D-quadratic or algebraic curve can be converted to the discrete connected listSIM-curve which can be integrated into rigid simplified CNC machine tools; the listSIM-curve can also be used as a benchmark for every IPO [1; 2]. The listing of listSIM of the Trident-curve [1; 2] compares with [15 Fig.4], therefore the result of the RMD-Trident-IPO compares with the adaptive ADA sampling algorithm of [15, §3.1]. The constant feedrate RMD-curves are more powerful and much simpler than the Pythagorean hodograph curves [16]. The Pythagorean hodograph curves have the nice property of polynomial dependence of arc length on curve parameter [15] and the constant feedrate listSIM-curves are discrete dependent on the rounded arc length . (Table 1) compares some properties of the PH and RMD curves.

Pythagorean-hodograph (PH) curves	RMD curves
Simple principle but extremely complicated implementations.	The principles are mostly forgotten and therefore extremely difficult to comprehend, but the real-time implementations are simple and the listSIM-arrays $\{\text{nputs}[i*T_c\,], \mathbf{P}[i*T_c\,], \mathbf{G}P[i*T_c\,]\}\big\|_{i=1}^{i=nkeyM}$ can be used as the real-time constant feedrate algorithms or as the benchmark of all real-time curve algorithms.
PH method started in 1990 and has fascinated many researchers and one can find hundreds of papers devoted to this topic	The RMD method started in 2015 and only five papers were devoted to this group of reference pulse integrators.
The PH method is valid for sampled-data systems.	The RMD-interpolators can replace the sampled-data interpolators [1, §1.2] and the 2D-examples [1; 2, §8] and the 3D-examples a, b, c, d, e, f [4] show that the execution time is excellent.

Table 1. Comparison of PH- and RMD-curves

The RMD method can be grouped into the “Reference Pulse Interpolators” and these reference pulse interpolators, based on the book [19] of 2008 have still a bad name [18] : “(Reference Pulse Interpolators) can achieve high accuracy but high-speed machining cannot be obtained, and high-performance CPU is required.” The big problems of these reference pulse interpolators was the grid-distance criterion [1; 2, §1.1], the non-monotonic approach and the non-constant feedrate. Therefore, it is unsuitable to group the RMD interpolators as reference pulse interpolators. But the postulate of [18] is false : all well-designed reference pulse interpolators can even be implemented with the low-cost 32-bit Highest-Performance microprocessors with FPU of Microchip. The Ducomat-CNC [11; 2, §1.1] (a real reference pulse interpolator system) used a PDP-8E computer of the years 1970-1975 without floating-

point hardware; the performance of this double word 24-bit computer is lower than the performance of the contemporary 32-bit microprocessors.

The study of the analytic geometry of general conics, has gone in and out of fashion several times over the past century and especially "polars (lines) and poles are advanced topics." Salmon [7] gives at page 91, Art. 98 the switching property or La Hire's Theorem (3A):" if a point $\mathbf{P}_2$ lies on the polar (line) of $\mathbf{P}_1$, then $\mathbf{P}_1$ lies on the polar (line) on $\mathbf{P}_2$ ". With this theorem you can prove the incremental equation of general conics (4A), and the relation (5A) between the twopoint decision and the midpoint decision variable for quadratic equations.

The RMD is based on three forgotten items of the Analytic Geometry

1) La Hire's Theorem (3A),
2) The Incremental equation (4A),
3) The relationship between the twopoint and the midpoint or better the arithmetic mean residue (5A) and the residue in the midpoint (6A),
4) The link between the Relative Minimum Distance to the circle and the Relative Minimum Distance to the polar line of the circle (33).

The first midpoint and twopoint algorithms were not based on the items 1, 2, 3 and 4, although they updated their equations without referencing the incremental equation (4A).

Donald L. Vossler [8], a mechanical engineer and computer software designer with more than 20 years' experience in computer aided design and geometric modelling, gives in [8, chapter 18.2] many theorems concerning poles and polars. He knew La Hire's theorem but not the items 2, 3 and 4.

Salmon proves many theorems using "circles" and then says that these theorems are true of all curves of the second degree. Jack Bresenham advised me to convert [5] to the circle $F[x,y] = x^2+y^2-R^2$ and this makes the construction of the pole $\mathbf{P}_E$ [5, page 24] doable, such that the midpoint OoA-parameter τ_E [5, (43)] and the midpoint correction function λ_{mE} [5, (70)] are exactly known, therefore the "Relative Minimum Distance Theorem" [5, page 27]] becomes clearer (Section 1.5).

The two extreme points $\{\mathbf{P}_{ext1}, \mathbf{P}_{ext2}\}$, define the monotonic direction $\mathbf{T}_{md} \triangleq \mathbf{P}_{ext2} - \mathbf{P}_{ext1}$, and the candidate points using the monotonic signs $\{Sx, Sy\} \triangleq Sign[\mathbf{T}_{md}]$.

MaxErr [1; 2; Table 1] needs the Euclidean distance to the monotonic segment of the quadratic curve or the implicit algebraic curve.

Taubin [9] sought new algorithms for rasterizing implicit curves. Taubin's first approximation to the true geometric distance $d[|\boldsymbol{\rho}_n|] = \dfrac{|F[x_n, y_n]|}{|G[x_n, y_n]|}$ cannot be used to measure MaxErr and cannot be used to select the best candidate point of $\{\mathbf{P}_u, \mathbf{P}_v\}$ to the curve.

The father of the *midpoint technique* is Van Aken [14], although Pitteway [13] published the midpoint method in 1967 and generated an 8-connected conic and he claimed that it was optimal[1] " … each incremental move being chosen by the computer to minimize the displacement from the intended curve." His article received little attention until Van Aken and Novak [14] published their midpoint algorithms for simple conics and compared their midpoint method with the twopoint method of Bresenham. They call the arithmetic mean method the twopoint method [14, pp 150] and they wrongly rejected the twopoint method by stating [14, pp 150]: "… the fact that the error terms F_B and F_D are nonlinear makes them unreliable indicators of linear error." The equations (5A) and (6A) disprove that statement.

The feedrate of lines and circles was not constant and the offset could rather easily be calculated for lines-circles-lines, but not for general conics. Huypens [5] extended Bresenham's algorithms to conics and explained the inherent errors OoC (Out of Control) and

[1] The proof was wrong

demystified the enigmatic OoA (Out of Accuracy). The monotonic direction prevents and detects OoC easily, and the next sections show that OoA is not a problem, although it was a problem during the last 50 years.

1.1 The LSD-algorithm and MaxErr

The LSD-algorithms [9, § 3-3.2] calculate offline (not in real time) the minimal (absolute) distance of the selected point $\mathbf{P}_n = \{x_n, y_n\}$ to an implicit polynomial curve F[x,y] = 0 . The vector distance of the point $\mathbf{P}_n$ to the curve F[x,y] = 0 is $\boldsymbol{\rho}_n \equiv \mathbf{P}_n - \mathbf{P}_{Fn}$ with $\mathbf{P}_{Fn} = \{x_{Fn}, y_{Fn}\}$ the footpoint of point $\mathbf{P}_n$ on the curve F[x,y] = 0.

a. The tangent in the footpoint $\mathbf{P}_{Fn} = \{x_{Fn}, y_{Fn}\} \in \mathbb{R}$ is

$\mathbf{T}_F[x_{Fn}, y_{Fn}] = \frac{1}{2}\{\frac{\partial F[x_{Fn}, y_{Fn}]}{\partial y}, \frac{-\partial F[x_{Fn}, y_{Fn}]}{\partial x}\}$ with the condition that the distance $\boldsymbol{\rho}_n$ is perpendicular to the tangent $\mathbf{T}_F[x_{Fn}, y_{Fn}]$. Hence the footpoint $\mathbf{P}_{Fn}$ is the solution of

$$\text{NSolve}\left[\left\{F[x_{Fn}, y_{Fn}] == 0, (x_n - x_{Fn}) * \frac{\partial F[x_{Fn}, y_{Fn}]}{\partial y} - (y_n - y_{Fn}) * \frac{\partial F[x_{Fn}, y_{Fn}]}{\partial x} == 0\right\}, \{x_{Fn}, y_{Fn}\}, \text{Reals}\right],$$

To avoid OoC, the direction in $\mathbf{P}_{Fn}$ must be conform to the sense of the monotonic direction, and the final solution $\mathbf{P}_{Fn}$ is the closest solution to $\mathbf{P}_n$.

The absolute minimum distance to the curve equals dErr[$\mathbf{P}_n$_] := N[Norm[$\mathbf{P}_n - \mathbf{P}_{Fn}$]].

b. Using the RegionDistance of Mathematica:
Define the ImplicitRegion as $\Re =$ ImplicitRegion [FP[**P**] == 0, {x,y}],
Define the absolute minimum distance to the curve as
dErr[$\mathbf{P}_n$_] := N[RegionDistance[$\Re, \mathbf{P}_n$] (always $\geq$ 0).

The computation of dErr[$\mathbf{P}_n$] is expensive and very time consuming, the calculation of the exact Euclidean distance, in real time, is impractical.

The Virtual LSD-algorithm and the RMD algorithm call the LSD-algorithm to compute the error dErr[$\mathbf{P}_n$] of the best selected point and put it in the array listErr={dErr[$\mathbf{P}_n$] }. The maximum error MaxErr equals Max[listErr].

The LSD-algorithm has no OoA errors and its MaxErr is minimal provided that the calling program makes no selection errors.

The selection rule of the Virtual LSD-algorithm or the RMD algorithm, for two points { $\mathbf{P}_B$, $\mathbf{P}_D$ } near the curve is "Select point $\mathbf{P}_B$ if Sign[$\boldsymbol{\rho}_B^2 - \boldsymbol{\rho}_D^2$] < 0, else select point $\mathbf{P}_D$."

1.2 Simplification for the circle $\mathbf{F[x,y] = x^2 + y^2 - R^2}$ (Fig. 2, Fig. 3)

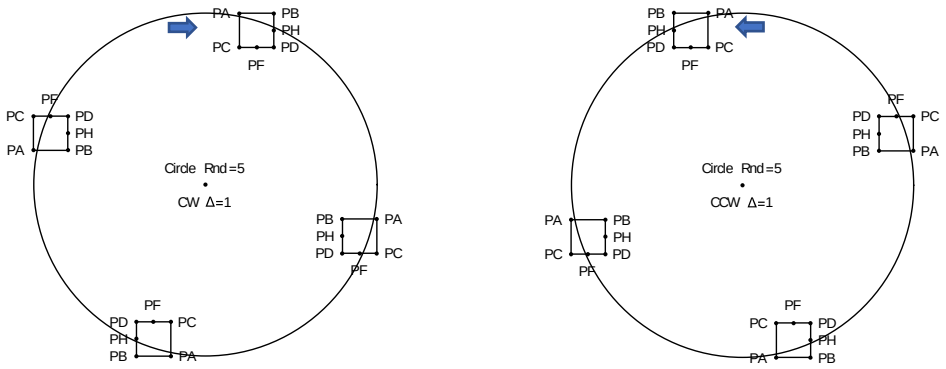


Fig. 3 Eight-Connected Candidate points with a high change of an OoA error

For the circle (CW) FP[**P**]≡ F[x, y] = $x^2 + y^2 - R^2$ (Eq 1), the gradient **G**P**[P]** ≡ **G**[x,y] $\triangleq \frac{1}{2}$Grad[F[x, y],{x, y}] (Eq 2) in the point **P** = {x,y}, simplifies to the point **P**. The tangent in **P** in CW-direction is **T**[**P**] ≡ -Cross[**G**P[**P**]] ≡{y, -x} (Eq 3), the extreme monotonic points are respectively { {0, R}, {R, 0}, {0, -R}, {-R,0}, {0, R} }. For the extreme points { $\mathbf{P}_{ext1}$, $\mathbf{P}_{ext2}$ }= {0,R}, {R,0}, the monotonic direction equals $\mathbf{T}_{md} \triangleq \mathbf{P}_{ext2} - \mathbf{P}_{ext1} \triangleq$ {R, -R} (Eq 4), the monotonic signs {Sx, Sy}$\triangleq$ Sign[$\mathbf{T}_{md}$] equal {1, -1} (Eq 5), the first best point is {0,Rnd} and the next best point is the point $\mathbf{P}_A = \{Sx, Rnd\}$ with Rnd = Round[R] (Fig. 3).

This paper gives the results for Rnd equal to 5, 50, 100, 400, and Rnd equal to infinity reduces the circle to a linear line. As stated by Salmon, the conclusion of the results are general.

The candidate points of the best point $\mathbf{P}_A$ are respectively { $\mathbf{P}_B, \mathbf{P}_D, \mathbf{P}_C$ } $\triangleq$ { $\mathbf{P}_A$ +{Sx, 0}, $\mathbf{P}_A$ +{Sx, Sy}, $\mathbf{P}_A$ +{0, Sy}} (Eq 6). The midpoints in the major x- and y-direction are { $\mathbf{P}_H, \mathbf{P}_V$ } = $\{\{x_H, y_H\}, \{x_V, y_V\}\}$ = { $\frac{\mathbf{P}_B + \mathbf{P}_D}{2}, \frac{\mathbf{P}_C + \mathbf{P}_D}{2}$ }={{2, Rnd-0.5}, {1.5, Rnd-1}} (Eq 7).

The twopoint algorithm [11] and the midpoint algorithm [10; 13; 14] were ideal suited to select the best candidate point of { $\mathbf{P}_B, \mathbf{P}_D$ } without calculating the exact distance of $\mathbf{P}_B$ or $\mathbf{P}_D$ to the circle. The vector distances of the candidate points { $\mathbf{P}_B, \mathbf{P}_D$ } to the circle are respectively $\{\boldsymbol{\rho}_B, \boldsymbol{\rho}_D\} = \{\{x_{\boldsymbol{\rho}_B}, y_{\boldsymbol{\rho}_B}\}, \{x_{\boldsymbol{\rho}_D}, y_{\boldsymbol{\rho}_D}\}\} = \{\mathbf{P}_B - \mathbf{P}_{FB}, \mathbf{P}_D - \mathbf{P}_{FD}\}$ (Eq 8) or $\{\boldsymbol{\rho}_B, \boldsymbol{\rho}_D\} = \{(1 - \frac{R}{Norm[\mathbf{P}_B]})\mathbf{P}_B, (1 - \frac{R}{Norm[\mathbf{P}_D]})\mathbf{P}_D\}$ (Eq 8), with respectively $\mathbf{P}_{FB} = \{x_{FB}, y_{FB}\} = \frac{R}{Norm[\mathbf{P}_B]}\mathbf{P}_B$ (Eq 9) and $\mathbf{P}_{FD} = \{x_{FD}, y_{FD}\} = \frac{R}{Norm[\mathbf{P}_D]}\mathbf{P}_D$ (Eq 10) the footpoints of the candidate points { $\mathbf{P}_B, \mathbf{P}_D$ } on the circle. The midpoint (old notation $\mathbf{P}_{FH}$) $\mathbf{P}_{mF} = \{x_{mF}, y_{mF}\} \triangleq \frac{\mathbf{P}_{FB} + \mathbf{P}_{FD}}{2}$ (Eq 11) is important for the construction of the pole $\mathbf{P}_E$ (§ 1.3.2).

The RMD algorithms calculate the best point or the Relative Minimum Squared Distance from the candidate points to the curve. Therefore, the RMD algorithms order the candidate points along decreasing priority in the array listCan, which equals $\{\mathbf{P}_D, \mathbf{P}_B, \mathbf{P}_C\}$ for a 2D-curve.

The Virtual LSD-algorithm can compute the absolute distances of all points in listCan and choose the point with the minimum distance to the curve; or it can use a <u>pairwise comparison algorithm</u>,

- Put $\mathbf{P}_u$ equal to $\mathbf{P}_D$,
- Compute the absolute distance $d_u = \|\boldsymbol{\rho}_C[\mathbf{P}_u]\|$ of $\mathbf{P}_u$,
- Compute the distance $d_B = \|\boldsymbol{\rho}_C[\mathbf{P}_B]\|$ and apply $\mathbf{P}_u = If[d_u - d_B] \le 0, \mathbf{P}_u, \{\mathbf{P}_B, d_u = d_B\}]$,
- Compute the absolute distance $\rho_C[\mathbf{P}_C]$ and apply $\mathbf{P}_u = If[d_u - d_C] \le 0, \mathbf{P}_u, \{\mathbf{P}_C, d_u = d_C]$,
- The point $\mathbf{P}_u$ has the minimum absolute distance d_u.

The pairwise comparison algorithm is not essential. You may say that the pairwise comparison algorithm measures <u>two</u> distance functions relative to each other or that it measures the difference of <u>two</u> distance functions relative to zero.

Adding the word “linear” means that the functions become linear distance functions as “distance to curve => distance to line”.

Adding the word “relative” means that the difference of two distance functions reduces to <u>a</u> decision function in the next form, which reduces to a scalar form.

"Sign[$\rho_u - \rho_v$]≡Sign[Norm[$\boldsymbol{\rho}_u$] -Norm[$\boldsymbol{\rho}_v$]]=>Sign[$\boldsymbol{\rho}_u \bullet \boldsymbol{\rho}_u - \boldsymbol{\rho}_v \bullet \boldsymbol{\rho}_v$] =Sign[$(\boldsymbol{\rho}_u - \boldsymbol{\rho}_v).(\boldsymbol{\rho}_u + \boldsymbol{\rho}_v)$]".
The relative linear decision functions (39) and (40) are truly relative linear forms.
The RMD-criterium measures, in essence, the sign of the linear decision function relative to zero. Therefore, we renamed the "Relative Curve Measurement Theorem" as "the "Relative Minimum Linear Distance Curve Measurement" and the abbreviation RMLD would be better than RMD.
All moves for 8-connected moves are major axis moves, that is not the case for 26-connected curves [4, § 2.5] and therefore, the selection rule for 26-connected moves is different [4, § 2.7]. When we apply that rule to 2D-curves, the ordered candidate points along decreasing priority become listCan $\equiv \{\mathbf{P}_D, \mathbf{P}_B, \mathbf{P}_C\}$. The RMD algorithm computes the decision function $d[\mathbf{P}_u, \mathbf{P}_v] \equiv d_M[\mathbf{P}_u, \mathbf{P}_v]$ (39) for the midpoint algorithm or $d[\mathbf{P}_u, \mathbf{P}_v] \equiv d_2[\mathbf{P}_u, \mathbf{P}_v]$ (40) for the twopoint algorithm and applies the RMD selection algorithm,

- Put $\mathbf{P}_u$ equal to $\mathbf{P}_D$,
- Apply $\mathbf{P}_u = \text{If}[d[\mathbf{P}_u, \mathbf{P}_B] \le 0, \mathbf{P}_u, \mathbf{P}_B]$,
- Apply $\mathbf{P}_u = \text{If}[d[\mathbf{P}_u, \mathbf{P}_C] \le 0, \mathbf{P}_u, \mathbf{P}_C]$,
- The point $\mathbf{P}_u$ has the unknown minimum absolute distance.

When the RMD algorithm has no OoA errors, the RMD algorithm and the VIRTUAL LSD algorithm have the same best point $\mathbf{P}_u$*, but the RMD never calculates the truly distance. Therefore, the linear RMD algorithm is many, many times faster than the VIRTUAL LSD algorithm.*
Huypens used the term "OoA" in 2015 [5], and instead of "OoA", he used the term "Out of Tolerance" in 2008 [6]. Many counterexamples showed that either the midpoint or the twopoint method was wrong (OoA). Sometimes both became berserk (OoC) because these algorithms did not use monotonic segments (the wright cause of OoC was not given [1; 2 "aliasing"]) . Therefore, the midpoint and twopoint algorithms got a bad name. Moreover, the feedrate was not constant and the generation of the offset curve was only successful for the line-circle-line combinations.
Huypens [5, page 32] located the OoA-segment: the midpoint $\mathbf{P}_H$ is "inside" the conic and extremely near to the conic edge. Therefore, the critical radii equal Norm[$\mathbf{P}_H$] ≡ $\sqrt{2^2 + (\text{Rnd} - 0.5)^2}$ for Rnd ={ 5, 50, 100, 400}. For Rnd, the radius R of the circle belongs to the range Rnd-0.5 < R < Rnd+0.5. So, for Rnd equal to 5, the critical R equals 4.92443 and for R equal to 4.925 the midpoint $\mathbf{P}_H$ is inside and close to the circle edge and the midpoint algorithm for R equal to 4.925 will probable make an OoA error. Bresenham presented a typical OoA error (Table 2) and his example [12, page 202] shows that sometimes the twopoint method is better than the midpoint method.
Each Rnd is further subdivided into seven subranges such that Rnd-0.5 < R < Rnd+0.5.
The R's are listed in listRndR for each subrange of Rnd and selected with R = list*Rnd*R[[i]], i = {1, 2, 3, 4, 5, 6, 7}. The R's for i equal to 1 or 7 have sometimes a strange behavior; when we do not consider 1 or 7, then the subrange is "closed".
(Table 2) calculates the OoA errors :
list5R = {4.51, 4.92, 4.924, 4.925, 4.942, 4.95, 5.49},
critical R = 4.92443,
This subrange has an OoA midpoint error for R equal to 4.925,
This subrange has an OoA twopoint error for R equal to 4.93 and 4.942,
list50R = {49.51, 49.53, 49.54, 49.541, 49.56, 49.57, 50.49},
critical R = 49.5404,
This subrange has an OoA twopoint error for R equal to 49.541,
list100R = {99.51, 99.52, 99.521, 99.525, 99.53, 99.54, 100.49}.

critical R = 99.5201,
This subrange has an OoA twopoint error for R equal to 99.521,
list400R = {399.501, 399.505, 399.506, 399.507, 399.508, 399.51, 400.499},
critical R = 399.505,
This subrange has no OoA errors,
list1000R = {999.501, 999.502, 999.503, 999.504, 999.505, 999.51, 1000.499},
critical R = 999.502,
This subrange has no OoA errors.

Circle FP[**P**]≡F[x,y]≡x²+y²-R², Clockwise, {Sx,Sy}={1,-1}, $\mathbf{P}_A$ ={1,5}, $\mathbf{P}_B$ ={2,5}, $\mathbf{P}_D$ ={2,4}

Rnd=Round[R]=5, $\mathbf{P}_H \triangleq \frac{\mathbf{P}_B + \mathbf{P}_D}{2}$, $2*(\mathbf{P}_B - \mathbf{P}_D)\bullet\mathbf{P}_H = 2*\{0,1\}\bullet\{2,4.5\} = F_B - F_D = 9$,

$F_P \equiv$ FP[**P**_]:=Chop[N[Dot[**P**,**P**]-R²]], $|\boldsymbol{\rho}_P| \equiv$ ρ[**P**_]:=Chop[N[$\sqrt{\text{Dot}[\mathbf{P},\mathbf{P}]}$ -R]],

GP[**P**_]:=0.5*Grad[F[x,y],{x,y}]={x,y}=**P**, $F_H \equiv FP[\mathbf{P}_H] = \text{Chop}[N[\text{Dot}[\mathbf{P}_H, \mathbf{P}_H] - R^2]]$,

$F_{aH} \triangleq \frac{F_B + F_D}{2}$, $\lambda_H = \frac{(\mathbf{P}_B - \mathbf{P}_D)\bullet(\mathbf{G}P[\mathbf{P}_B] - \mathbf{G}P[\mathbf{P}_D])}{4} = F_{aH} - F_H = 0.25$,

$\text{If}[|\boldsymbol{\rho}_B| < |\boldsymbol{\rho}_D|], \text{"Select } \mathbf{P}_B \text{"}, \text{"Select } \mathbf{P}_D \text{"}]$,

$\text{If}[F_H < 0, \text{"}F_H \text{ selects } \mathbf{P}_B \text{"}, \text{"}F_H \text{ selects } \mathbf{P}_D \text{"}]$,

$\text{If}[F_{aH} < 0, \text{"}F_{aH} \text{ selects } \mathbf{P}_B \text{"}, \text{"}F_{aH} \text{ selects } \mathbf{P}_D \text{"}]$,

If [! Equivalent[$|\boldsymbol{\rho}_B| < |\boldsymbol{\rho}_D|, F_H < 0$], "$F_H$ is OoA", "F_H is OK"],

If [! Equivalent[$|\boldsymbol{\rho}_B| < |\boldsymbol{\rho}_D|, F_{aH} < 0$], "$F_{aH}$ is OoA", "F_{aH} is OK"]

R	$F_B \simeq$	$F_D \simeq$	$F_{aH} \simeq$	$F_H \simeq$	$\|\boldsymbol{\rho}_B\|$	$\|\boldsymbol{\rho}_D\|$	OoA errors ?
4.925	4.744	-4.256	0.2444	-056	0.460	0.453	F_H selects $\mathbf{P}_B \Rightarrow F_H$ is OoA F_{aH} selects $\mathbf{P}_D \Rightarrow F_{aH}$ is OK Hence Midpoint is OoA
4.93	4.695	-4.305	0.1951	-0.0549	0.455	0.458	F_H selects $\mathbf{P}_B \Rightarrow F_H$ is OK F_{aH} selects $\mathbf{P}_D \Rightarrow F_{aH}$ is OoA Hence Twopoint is OoA

Table 2. Typical OoA error: Bresenham's example shows only the row for R equal to 4.925

1.3 The calculation of the pole $\mathbf{P}_E$

1.3.1 The new notation of the points of paper [5]

This paper redefines the midpoint variables of paper [5] and uses a new notation. Paper [5] used the midpoint $\mathbf{P}_M$ of $\mathbf{P}_B$ and $\mathbf{P}_C$, but this paper uses the midpoint $\mathbf{P}_H$ of ($\mathbf{P}_B$, $\mathbf{P}_D$). The new notations for the midpoints $\mathbf{P}_{FM}$ and $\mathbf{P}_{EM}$ are respectively $\mathbf{P}_{mF}$ and $\mathbf{P}_{mE}$. The points { $\mathbf{P}_{FB}$, $\mathbf{P}_{FD}$ } are respectively the footpoints of the respectively points { $\mathbf{P}_B$, $\mathbf{P}_D$ } on the circle and the tangent of the angle θ_{BF} of the line $\mathbf{P}_{FD} \mapsto \mathbf{P}_{FB}$ equals tgBF[R, Rnd] = -Tan[$\frac{y_{FB} - y_{FD}}{x_{FB} - x_{FD}}$] (Eq 12). The footpoints { $\mathbf{P}_{FB}$, $\mathbf{P}_{FD}$ } belong to the polar line $\mathbf{T}_{pF}$ and define the pole $\mathbf{P}_F$ (20). So, the pole of the polar line $\mathbf{T}_{pF}$ is the point $\mathbf{P}_F$ and the points { $\mathbf{P}_{FB}$, $\mathbf{P}_{FD}$ }

$\in \mathbf{T}_{pF} = \mathbf{P} \bullet \mathbf{P}_F - R^2$ are on the circle. The letter "F" refers to the index of the pole $\mathbf{P}_F$ and not to the first letter "F" of footpoint. The midpoint of the points $\{\mathbf{P}_{FB}, \mathbf{P}_{FD}\}$ is the point

$$\mathbf{P}_{mF} = \frac{\mathbf{P}_{FB} + \mathbf{P}_{FD}}{2} \quad (11)$$

1.3.2 The calculations of $\mathbf{P}_E$, $\{\mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ and $\mathbf{P}_{mE}$

The distance of the points $\{\mathbf{P}_B, \mathbf{P}_D\}$ to the circle are $\{\boldsymbol{\rho}_B, \boldsymbol{\rho}_D\} \equiv \{\mathbf{P}_B - \mathbf{P}_{FB}, \mathbf{P}_D - \mathbf{P}_{FD}\}$ (8) with $\{x_{\boldsymbol{\rho}_B}, y_{\boldsymbol{\rho}_B}\} \triangleq \boldsymbol{\rho}_B$ and $\{x_{\boldsymbol{\rho}_D}, y_{\boldsymbol{\rho}_D}\} \triangleq \boldsymbol{\rho}_D$.

The unit direction of the line segment $\mathbf{P}_{FB}$ - $\mathbf{P}_{FD}$ is $\overrightarrow{\text{DirBF}} \triangleq \{\text{Cos}[\theta_{BF}], \text{Sin}[\theta_{BF}]\}$.

The line $(y - y_{mF}) = \text{tgBF}[R, \text{Rnd}] * (x - x_{mF})$ is the same as the polar $\mathbf{T}_{pF}$ of the pole $\mathbf{P}_F$.

The pole of the polar line $\boldsymbol{T_{pE}}$ is the point $\boldsymbol{P_E}$:

To find $\mathbf{P}_{mE}$, the pole $\mathbf{P}_E$ and the intersections $\{\mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ of the polar line $\mathbf{T}_{pE}$ of the pole $\mathbf{P}_E$ with the circle $x^2 + y^2 - R^2 \equiv 0$, we define the unit direction of the line segment $\mathbf{P}_{EB}$ - $\mathbf{P}_{ED}$ as $\overrightarrow{\text{DirBE}} \triangleq \{\text{Cos}[\theta_{BE}], \text{Sin}[\theta_{BE}]\}$ (Eq 13). The points $\{\mathbf{P}_{EB}, \mathbf{P}_{ED}\} \in \mathbf{T}_{pE} = \mathbf{P} \bullet \mathbf{P}_E - R^2$ are on the circle and the letter "E" refers to the index of the pole $\mathbf{P}_E$. The tangent of the angle θ_{BE} of the line $\mathbf{P}_{ED} \mapsto \mathbf{P}_{mF} \mapsto \mathbf{P}_{EB}$ equals $\text{tgBE}[R, \text{Rnd}] = \text{Tan}[\theta_{BE}]$ (14).

The points $\{\mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ are the intersections of the line $\mathbf{T}_{pE}$, rotated around the pivoting point $\mathbf{P}_{mF}$ (11), with the circle such that $(\boldsymbol{\rho}_B \bullet \mathbf{T}_{pE})^2 \equiv (\boldsymbol{\rho}_D \bullet \mathbf{T}_{pE})^2$ [5, (27)] , [5, Construction of the pole $\mathbf{P}_E$, p.23] or $(\boldsymbol{\rho}_B \bullet \overrightarrow{\text{DirBE}})^2 \equiv (\boldsymbol{\rho}_D \bullet \overrightarrow{\text{DirBE}})^2$. The solutions of $\overrightarrow{\text{DirBE}}$ is

$\text{Solve}[(\boldsymbol{\rho}_B \bullet \{\text{Cos}[\theta_{BE}], \text{Sin}[\theta_{BE}]\})^2 == (\boldsymbol{\rho}_D \bullet \{\text{Cos}[\theta_{BE}], \text{Sin}[\theta_{BE}]\})^2,$

$\text{Cos}^2[\theta_{BE}] + \text{Sin}^2[\theta_{BE}] == 1\}, \{\text{Cos}[\theta_{BE}], \text{Sin}[\theta_{BE}]\}, \text{Reals}]$,

and the accepted solution is $\text{tgBE}[R, \text{Rnd}] = \dfrac{\text{Sin}[\theta_{BE}]}{\text{Cos}[\theta_{BE}]} = -\text{Tan}[\dfrac{x_{\boldsymbol{\rho}_B} - x_{\boldsymbol{\rho}_D}}{y_{\boldsymbol{\rho}_B} - y_{\boldsymbol{\rho}_D}}]$ (Eq 14).

The polar $\mathbf{T}_{pE}$ of the pole $\mathbf{P}_E$ is the line $(y - y_{mF}) = \text{tgBE}[R, \text{Rnd}] * (x - x_{mF})$ (Eq 15).

Mathematica cannot find the solution of the intersection of that line with the circle (1), but using the new variables TGBE and $\{x_{MF}, y_{MF}\}$ instead of the known functions tgBE (14) and $\{x_{mF}, y_{mF}\}$ (11) gives

$$(x)_{EB} = -\frac{1}{1 + \text{TGBE}^2}(-\text{TGBE}^2 * x_{MF} + \text{TGBE} * y_{MF} + \sqrt{R1^2 * (1 + \text{TGBE}^2) - (y_{MF} - \text{TGBE} * x_{MF})^2}\ )$$

$$(y)_{EB} = \frac{1}{1 + \text{TGBE}^2}(y_{MF} - \text{TGBE}(x_{MF} + \sqrt{R1^2 (1 + \text{TGBE}^2) - (y_{MF} - \text{TGBE} * x_{MF})^2}\ )$$

$$(x)_{ED} = \frac{1}{1 + \text{TGBE}^2}(\text{TGBE}^2 * x_{MF} - \text{TGBE} * y_{MF} + \sqrt{R1^2 * (1 + \text{TGBE}^2) - (y_{MF} - \text{TGBE} * x_{MF})^2}\ )$$

$$(y)_{ED} = \frac{1}{1 + \text{TGBE}^2}(y_{MF} + \text{TGBE}(-x_{MF} + \sqrt{R1^2 (1 + \text{TGBE}^2) - (y_{MF} - \text{TGBE} * x_{MF})^2}\ ) .$$

The "ReplaceAll" function /. listREP_{EB}, with

$\text{listREP}_{EB} = \{\text{TGBE} \to \text{TgBE}[R, \text{Rnd}], x_{MF} \to x_{mF}[R, \text{Rnd}], y_{MF} \to y_{mF}[R, \text{Rnd}]\}$ gives

$\mathbf{P}_{EB}=\{x_{EB}, y_{EB}\}=\{(x)_{EB}, (y)_{EB}\}$ /. listREP_{EB}, (Eq 16)

$\mathbf{P}_{ED}=\{x_{ED}, y_{ED}\}=\{(x)_{ED}, (y)_{ED}\}$ /. listREP_{EB}. (Eq 17)

The midpoint of the points $\{\mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ is the point $\mathbf{P}_{mE}=\{x_{mE}, y_{mE}\}=\frac{\mathbf{P}_{EB}+\mathbf{P}_{ED}}{2}$. (Eq 18)

$\mathbf{P}_E$ is the intersection of the polars in $\mathbf{P}_{EB}$ and $\mathbf{P}_{ED}$, hence

$$\mathbf{P}_E = \text{Flatten}[\text{LinearSolve}[\begin{pmatrix} x_{EB} & y_{EB} \\ x_{ED} & y_{ED} \end{pmatrix}, R^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}]] = \frac{R^2}{\text{Det}[\begin{pmatrix} x_{EB} & y_{EB} \\ x_{ED} & y_{ED} \end{pmatrix}]}\{y_{ED}-y_{EB}, x_{EB}-x_{ED}\}. \text{ (Eq 19)}$$

The "E" refers to the index of the pole $\mathbf{P}_E$ of the polar line $\{\mathbf{P}_{EB}, \mathbf{P}_{ED}\}$.

$\mathbf{P}_F$ is the intersection of the polars in $\mathbf{P}_{FB}$ and $\mathbf{P}_{FD}$, hence

$$\mathbf{P}_F = \frac{R^2}{\text{Det}[\begin{pmatrix} x_{FB} & y_{FB} \\ x_{FD} & y_{FD} \end{pmatrix}]}\{y_{FD}-y_{FB}, x_{FB}-x_{FD}\}. \text{ (Eq 20)}$$

Using (9) and (10) $\mathbf{P}_F$ becomes with

$$x_F[R, Rnd] = R * \frac{\sqrt{4+Rnd^2} - Rnd\left(\sqrt{4+Rnd^2} - \sqrt{4+(Rnd-1)^2}\right)}{2} \text{ (Eq 21),}$$

$$y_F[R, Rnd] = R * \left(\sqrt{4+Rnd^2} - \sqrt{4+(Rnd-1)^2}\right) \text{ (Eq 22),}$$

$$\mathbf{P}_F[R, Rnd] = \{x_F[R, Rnd], y_F[R, Rnd]\}. \text{ (Eq 23)}$$

All the points $\{\mathbf{P}_{FB}, \mathbf{P}_{FD}, \mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ are on the edge of the circle and the points $\{\mathbf{P}_{mF}, \mathbf{P}_{mE}\}$ are inside the circle. The poles $\{\mathbf{P}_F, \mathbf{P}_E\}$ are at the outside of the circle.

Paper [5] did not compute the values of $\{\mathbf{P}_E, \mathbf{P}_{mE}, \mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ and even for the simple circle the general formulae are too long to be given, therefore the paper calculated only the values for certain R's and Rnd's.

Using (9), (10) and (20), the points $\{\mathbf{P}_F, \mathbf{P}_{mF}, \mathbf{P}_{FB}, \mathbf{P}_{FD}\}$ can be calculated rather easily . (Table 3) shows that he points $\{\mathbf{P}_F, \mathbf{P}_{mF}, \mathbf{P}_{FB}, \mathbf{P}_{FD}\}$ are close to the points $\{\mathbf{P}_E, \mathbf{P}_{mE}, \mathbf{P}_{EB}, \mathbf{P}_{ED}\}$ (the greater the value of R, the closer the corresponding values !).

R	$\mathbf{P}_{EB}$	$\mathbf{P}_{ED}$	$\mathbf{P}_{mE}$	$\mathbf{P}_E$
4.51	{1.451, 4.2702}	{1.9224, 4.0798}	{1.6867, 4.175}	{1.6921, 4.1883}
4.92	{1.8236, 4.5696}	{2.1969, 4.4023}	{2.0102, 4.4859}	{2.0137, 4.4937}
4.924	{1.8268, 4.5726}	{2.2002, 4.4051}	{2.0135, 4.4888}	{2.017, 4.4966}
4.925	{1.8276, 4.5734}	{2.2011, 4.4058}	{2.0143, 4.4896}	{2.0178, 4.4973}
4.93	{1.8315, 4.5772}	{2.2053, 4.4093}	{2.0184, 4.4932}	{2.0219, 4.501}
4.942	{1.8409, 4.5863}	{2.2156, 4.4175}	{2.0282, 4.5019}	{2.0318, 4.5097}
5.49	{2.1748, 5.0409}	{2.8181, 4.7115}	{2.4965, 4.8762}	{2.5073, 4.8974}
99.521	{1.99, 99.5011}	{2.0101, 99.5007}	{201, 99.5009}	{201, 99.5009}
399.505	{1.9975, 399.5}	{225, 399.5}	{2., 399.5}	{2., 399.5}
R	$\mathbf{P}_{FB}$	$\mathbf{P}_{FD}$	$\mathbf{P}_{mF}$	$\mathbf{P}_F$
4.51	{1.675, 4.1874}	{2.0169, 4.0339}	{1.846, 4.1106}	{1.8491, 4.1178}
4.92	{1.8272, 4.5681}	{2.2003, 4.4006}	{2.0138, 4.4843}	{2.0173, 4.4921}
4.924	{1.8287, 4.5718}	{2.2021, 4.4042}	{2.0154, 4.488}	{2.0189, 4.4958}

4.925	{1.8291, 4.5727}	{2.2025, 4.4051}	{2.0158, 4.4889}	{2.0193, 4.4967}
4.93	{1.831, 4.5774}	{2.2048, 4.4095}	{2.0179, 4.4935}	{2.0214, 4.5012}
4.942	{1.8354, 4.5885}	{2.2101, 4.4203}	{2.0228, 4.5044}	{2.0263, 4.5122}
5.49	{2.0389, 5.0973}	{2.4552, 4.9104}	{2.2471, 539}	{2.251, 5.0125}
99.521	{1.99, 99.5011}	{2.0101, 99.5007}	{201, 99.5009}	{201, 99.5009}
399.505	{1.9975, 399.5}	{225, 399.5}	{2., 399.5}	{2., 399.5}

Table 3 The intersections of the polars of respectively the poles $\mathbf{P}_E$ and $\mathbf{P}_F$ with the circle

1.4 The right triangle altitude theorem $h_{mF}^2 = \text{Norm}^2[\mathbf{P}_{FB} - \mathbf{P}_{mF}] \equiv \lambda_{mF} \equiv \mathbf{P}_{mF} \bullet (\mathbf{P}_F - \mathbf{P}_{mF})$

Consider the right triangle $\{\mathbf{P}_0, \mathbf{P}_F, \mathbf{P}_{FB}\}$ with hypothenuse $\overbrace{\mathbf{P}_0\mathbf{P}_F}$ and altitude $h_{mF} \equiv \overbrace{\mathbf{P}_{BF}\mathbf{P}_{mF}}$.

The right triangle theorem gives the geometric meaning of the correction λ_{mF}.

The right triangle theorem states that $h_{mF}^2 \equiv \mathbf{P}_{mF} \bullet (\mathbf{P}_F - \mathbf{P}_{mF})$ (Eq 24) and

$\mathbf{P}_{BF} \bullet \mathbf{P}_{BF} \equiv \mathbf{P}_{mF} \bullet \mathbf{P}_F \equiv R^2$ (Eq 25). So, $h_{mF}^2 = R^2 - \mathbf{P}_{mF} \bullet \mathbf{P}_{mF} \triangleq -\text{FP}[\mathbf{P}_{mF}]$ (Eq 26). Otherwise, analogous to λ_{mE} from [5, (70)], $\lambda_{mF}[R, \text{Rnd}] = \dfrac{(\mathbf{P}_{FD} - \mathbf{P}_{FB}) \bullet (\mathbf{P}_{FD} - \mathbf{P}_{FB})}{4} = -\text{FP}[\mathbf{P}_{mF}]$ (Eq 27).

Hence, $h_{mF}^2 = \text{Norm}^2[\mathbf{P}_{FB} - \mathbf{P}_{mF}] \equiv \lambda_{mF}$ (Eq 28) or λ_{mF} is the square distance of the altitude $\overbrace{\mathbf{P}_{BF}\mathbf{P}_{mF}}$ of the right triangle $\{\mathbf{P}_0, \mathbf{P}_F, \mathbf{P}_{FB}\}$.

Analogously, the right triangle theorem is also valid for the right triangle $\{\mathbf{P}_0, \mathbf{P}_E, \mathbf{P}_{EB}\}$.

Therefore, all the equations {24, 25, 26, 27} are also valid for $\{\mathbf{P}_F, \mathbf{P}_{FB}, \mathbf{P}_{FD}, \mathbf{P}_{mF}, h_{mF}, \lambda_{mF}\}$ replaced with $\{\mathbf{P}_E, \mathbf{P}_{EB}, \mathbf{P}_{ED}, \mathbf{P}_{mE}, h_{mE}, \lambda_{mE}\}$ (Eq 29) .

(Table 4) shows that $R * \sqrt{\lambda_{mE}} = R * h_{mE} \cong R * \sqrt{\lambda_{mF}} = R * h_{mF} \simeq 1.$

This means that $h \triangleq h_{mE} \cong h_{mF} \simeq \dfrac{1}{R}$ (Eq 30) and that is only possible when $R^2 >> h^2$ or $R^4 \gg 1$.

Proof:

Let 2α equal the angle of $\measuredangle\ \mathbf{P}_{EB}\mathbf{P}_0\mathbf{P}_{ED}$, then

$$\text{Tan}[2\alpha] = \frac{2\,\text{Tan}[\alpha]}{1 - \text{Tan}^2[\alpha]} = \frac{2\frac{h}{R}}{\frac{R^2 - h^2}{R^2}} = \frac{2h * R}{R^2 - h^2} \simeq \frac{2}{R^2 - \frac{1}{R^2}} \simeq \frac{2 * R^2}{R^4 - 1} \simeq \frac{2}{R^2}\ .$$

So, the angle 2α is exceedingly small for $R^4 \gg 1$, therefore, $\alpha \simeq \dfrac{1}{R^2}$.

From (24), (28) and (27) $\|\mathbf{P}_{mE}\| \cong \|\mathbf{P}_{mF}\| \simeq R^2 - \dfrac{1}{R^2}$ (Eq 31).

From (26), (28) and (27) $\lambda_{mE} \cong \lambda_{mF} \simeq \dfrac{1}{R^2}$ (Eq 32).

The chord error r_h of the polar line $\boldsymbol{T}_{\boldsymbol{pE}}$:

The chord error r_h equals by definition $r_h \triangleq R - \|\mathbf{P}_{mE}\|$

For $h \simeq \dfrac{1}{R}$, the chord error r_h becomes

$$r_h = R - \sqrt{R^2 - \frac{1}{R^2}} = \frac{R^2 - \sqrt{R^4 - 1}}{R} = \frac{R^2 - R^2\left(1 - \frac{1}{2R^4}\right)}{R} = \frac{1}{2R^3}$$

So, the polar line $\mathbf{T}_{pE}$ of $\mathbf{P}_E$ with respect to the circle is almost the tangent of the circle.
Analogous, the polar line $\mathbf{T}_{pF}$ is almost the tangent of the circle.
As the circle is the envelope of the family of its tangent lines, the simple circle is *almost* the envelope of the polar lines $\mathbf{T}_{pE}$ of the RMD algorithm of the simple circle.
The distances of the candidate points to the circle equal the distances to the tangents of the circle and these distances *almost* equal the distances to the polar lines $\mathbf{T}_{pE}$. For $R^4 \gg 1$ the word *almost* becomes *always*.
There is no difference between the midpoint- and the twopoint RMD decision functions when Round[R] equals infinity or when the 2D-curve is a 2D-line. When (Round[radius of curvature])$^4 \gg 1$, the midpoint- and twopoint RMD decision functions measure *almost* the distance to the tangent of the envelope of the 2D-curve, as the curve is the envelope of the family of its tangent lines, and that is also valid for a 3D-curves.
Therefore and without rigorous mathematical proof the twopoint RMD draws transcendental equations, viz. y = Tanh[x]. With Δ equal to 0.01, and {Δx, Δy } = { Δ, Tanh[Δ] } ≅ { Δ , Δ }, the generated discrete curve is not 8-connected and therefore the twopoint RMD cannot apply the constant feedrate and the arclength algorithm for transcendental curves (see also section 3.1).

R = list5R ; critical R = 4.92443; within the closed subrange λ_{mE} approximates λ_{mF} and $R*\sqrt{\lambda_{mE}} \cong R*\sqrt{\lambda_{mF}} \simeq 1$

R	λ_{mE}	λ_{mF}	$\frac{1}{\sqrt{\lambda_{mE}}}$	$\frac{1}{\sqrt{\lambda_{mF}}}$	$R*\sqrt{\lambda_{mE}}$	$R*\sqrt{\lambda_{mF}}$
4.51	0.0646	0.0351	3.9335	5.3353	1.1466	0.8453
4.92	0.0418	0.0418	4.8899	4.8907	1.0062	1.006
4.924	0.0419	0.0419	4.8865	4.8868	1.0077	1.0076
4.925	0.0419	0.0419	4.8856	4.8858	1.0081	1.008
4.942	0.0422	0.0422	4.8669	4.869	1.0154	1.015
4.95	0.0424	0.0423	4.8558	4.8611	1.0194	1.0183
5.49	0.1306	0.0521	2.7676	4.3829	1.9837	1.2526
99.521	0.0001	0.0001	99.5165	99.5167	1.	1.
399.505	0.000006	0.000006	399.506	399.504	1.	1.

Table 4 $\lambda_{mE} \cong \lambda_{mF} \simeq \frac{1}{R^2}$ (32).

Hence, the OoA-correction for the curvature of a curve is simple.

1.5 The crux of the relative Curve Measurement Theorem of [5, §5]

The crux of the Relative Curve Measurement theorem states that the sign of the RMD to the Curve equals the sign of the RMD to the Polar Line .
The mathematical translation is

$$\text{Sign}[\overbrace{\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v]}^{\textbf{curve}}] \equiv \text{Sign}[(1-\varepsilon_E)*\mu_E*\tau_E]\overbrace{*\text{Sign}[(\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v])]}^{\text{polar } \textbf{line} \text{ of } \frac{\mathbf{P}_u+\mathbf{P}_v}{2}} \quad \text{(Eq 33)}$$

Proof:

$$\text{Sign}[\rho_C^2[\mathbf{P}_C]-\rho_C^2[\mathbf{P}_B]] = \text{Sign}[(1-\varepsilon_E)*\tau_E]*\text{Sign}[(\mathbf{P}_C-\mathbf{P}_B)\bullet\mathbf{GP}[\mathbf{P}_E]*\text{Sign}[F_M] \text{ [5, §5, (41)].}$$

Transforming $\{\mathbf{P}_B,\mathbf{P}_M,\mathbf{P}_C\}$ to $\{\mathbf{P}_B,\mathbf{P}_H,\mathbf{P}_D\}$ gives

$$\text{Sign}[\rho_C^2[\mathbf{P}_D]-\rho_C^2[\mathbf{P}_B]] = \text{Sign}[(1-\varepsilon_E)*\tau_E]*\text{Sign}[(\mathbf{P}_D-\mathbf{P}_B)\bullet\mathbf{GP}[\mathbf{P}_E]*\text{Sign}[\text{FP}[\mathbf{P}_H]].$$

Defining $\mu_E*(\mathbf{P}_D-\mathbf{P}_B)\bullet\mathbf{GP}[\mathbf{P}_H] \triangleq (\mathbf{P}_D-\mathbf{P}_B)\bullet\mathbf{GP}[\mathbf{P}_E]$ (Eq 34) and applying (9A) gives

Clarification of the Relative Minimum Distance (RMD) Theorems and the real-time arclength algorithm, DOI: 10.13140/RG.2.2.36835.84009

$$\text{Sign}[\overbrace{\rho_C^2[\mathbf{P}_D]-\rho_C^2[\mathbf{P}_B]}^{\textbf{curve}}] \equiv \text{Sign}[(1-\varepsilon_E)*\mu_E*\tau_E]*\overbrace{\text{Sign}[(\rho_L^2[\mathbf{P}_D]-\rho_L^2[\mathbf{P}_B])]}^{\text{polar } \textbf{line} \text{ of } \mathbf{P}_H}. \text{ (Eq 35)}$$

Transforming $\{\mathbf{P}_B, \mathbf{P}_H, \mathbf{P}_D\}$ to $\{\mathbf{P}_u, \frac{\mathbf{P}_u+\mathbf{P}_v}{2}, \mathbf{P}_v\}$ gives (33).

This paper clarifies the Relative Curve Distance Theorem for the simple circle (1).
The RMD algorithm uses the same selection rule as the VIRTUAL LSD-algorithm and applies the Relative Curve Measurement Theorem on the same selection rule. Therefore, the selection rule, for two points $\{\mathbf{P}_B, \mathbf{P}_D\}$ near the curve, simplifies to "Select point $\mathbf{P}_D$ if $\text{Sign}[\boldsymbol{\rho}_L^2[\mathbf{P}_D]-\boldsymbol{\rho}_L^2[\mathbf{P}_B]] \leq 0$, else select point $\mathbf{P}_B$", with $\{\boldsymbol{\rho}_L[\mathbf{P}_B], \boldsymbol{\rho}_L[\mathbf{P}_B]\}$ respectively the vector distances of the points $\{\mathbf{P}_B, \mathbf{P}_D\}$ to the Polar Line of the midpoint $\mathbf{P}_H$ of the points $\{\mathbf{P}_B, \mathbf{P}_D\}$.
Paper [5] proves the theorem for conics, papers [1; 2] extend it to 3D-lines and papers [3; 4] extend it to quadrics and QSICs and their examples show that it is valid for implicit algebraic curves. The theorem is valid for algebraic curves because in the region $\{\mathbf{P}_{FB}, \mathbf{P}_{mF}, \mathbf{P}_{FD}\}$, the algebraic curve can be approximated by a quadratic curve. Applying "The "Relative Curve Measurement" theorem to this quadratic segment also gives (35).
The RMD algorithms use the gradient of the curve (viz. for the computation of the constant feedrate, the offset, the midpoint criterium applying (1A), (2A), (3A), (4A), etc.). The gradients make the RMD algorithms more general and especially more flexible.
When we rewrite equation (35) using (10A), we get

$$\text{Sign}[\overbrace{(\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v])}^{\text{curve}}] \equiv \text{Sign}[(1-\varepsilon_E)*\mu_E*\tau_E]*\overbrace{\text{Sign}[(FP[\mathbf{P}_u]-FP[\mathbf{P}_v])*F_H]}^{\text{midpoint criterium}} \text{ (Eq 36).}$$

This paper extends the relative curve distance measurement to the twopoint criterium using a new parameter τ_2 such that $\tau_2 * F_{aH} \triangleq \tau_E * F_H$. (Eq 37). For quadratic and 3rd order algebraic equations the arithmetic mean F_{aH} equals $F_{aH} = F_H + \frac{(\mathbf{P}_D-\mathbf{P}_B)\bullet(\mathbf{G}P[\mathbf{P}_D]-\mathbf{G}P[\mathbf{P}_B])}{4}$ (5A).
For quadratic equations the difference of F_{aH} and F_H equals F_{aH} - $F_H = \lambda_H$ (5A; 6A).
When we rewrite the second part of the equations (36) using (37), we get

$$\text{Sign}[\overbrace{(\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v])}^{\text{curve}}] \equiv \text{Sign}[(1-\varepsilon_E)*\mu_E*\tau_2]*\overbrace{\text{Sign}[(FP[\mathbf{P}_u]-FP[\mathbf{P}_v])*F_{aH}]}^{\text{twopoint criterium}} \text{ (Eq 38).}$$

(Tables 7, 8 and 9) show that $\text{Sign}[1-\varepsilon_E]$ and $\text{Sign}[\mu_E]$ are always one, therefore, we define the decision functions as

$$d_M[\mathbf{P}_u, \mathbf{P}_v] \triangleq \text{Sign}[\tau_E]*\overbrace{\text{Sign}[(FP[\mathbf{P}_u]-FP[\mathbf{P}_v])*F_H]}^{\text{midpoint criterium}} \text{ (Eq 39),}$$

$$d_2[\mathbf{P}_u, \mathbf{P}_v] \triangleq \text{Sign}[\tau_2]*\overbrace{\text{Sign}[(FP[\mathbf{P}_u]-FP[\mathbf{P}_v])*F_{aH}]}^{\text{two-point criterium}} \text{ (Eq 40),}$$

such that the difference of the squared distances become

$$\text{Sign}\left[\overbrace{\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v]}^{\text{curve}}\right] \equiv \text{Sign}\left[d_M[\mathbf{P}_u, \mathbf{P}_v]\right] \text{ (39),}$$

$$\text{Sign}\left[\overbrace{\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v]}^{\text{curve}}\right] \equiv \text{Sign}\left[d_2[\mathbf{P}_u, \mathbf{P}_v]\right] \text{ (40).}$$

1.5.1 The parameters of the RMD theorem (Table 5)

The "Relative Curve Measurement" Theorem of [5, § 5] defined the parameters ε_E [5, (42)] and τ_E [5, (43)] for the points $\{\mathbf{P}_B, \mathbf{P}_M, \mathbf{P}_C\}$. This paper transforms all parameters for the points $\{\mathbf{P}_B, \mathbf{P}_H, \mathbf{P}_D\}$ and (Table 5) defines the new parameters. The new parameter μ_E generalizes the theorem to (33) and the new parameter τ_2 extends the theorem to the twopoint criterium.

The theorem reduces to the simple midpoint criterium (39) or the simple twopoint criterium (40). (Table 3) and (Table 6, 7, and 8) prove that the approximations of (Table 5) are valid.

Parameters of the RMD theorem with $\mathbf{P}_{index} \triangleq \mathbf{G}_{index} \triangleq \mathbf{GP}[\mathbf{P}_{index}] \,\&\, \lambda_H = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet(\mathbf{G}_D - \mathbf{G}_B)}{4} = \frac{\Delta^2}{4}$			
With respect to the point $\mathbf{P}_E$		With respect to the point $\mathbf{P}_F$	
	almost		almost
$\tau_E = \frac{\mathbf{P}_E\bullet\mathbf{G}_H + W_H}{F_H} = \frac{\mathbf{P}_E\bullet\mathbf{P}_H - R^2}{\mathbf{P}_H\bullet\mathbf{P}_H - R^2}$	$\tau_E \to \tau_F$	$\tau_F = \frac{\mathbf{P}_F\bullet\mathbf{G}_H + W_H}{F_H} = \frac{\mathbf{P}_F\bullet\mathbf{P}_H - R^2}{\mathbf{P}_H\bullet\mathbf{P}_H - R^2}$	
$\varepsilon_E = \frac{(\mathbf{P}_{FD} - \mathbf{P}_{FB})\bullet\mathbf{G}_E}{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}_E}$	$1\text{-}\varepsilon_E \to 1$	$\varepsilon_F = \frac{(\mathbf{P}_{FD} - \mathbf{P}_{FB})\bullet\mathbf{G}_F}{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}_F}$	$\varepsilon_F \equiv 0$
$\mu_E = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}_E}{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}_H}$	$\mu_E \to 1$	$\mu_F = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}_F}{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}_H}$	$\mu_F \to 1$
$\tau_2 = \tau_E \frac{F_H}{F_{aH}} == \frac{\mathbf{P}_E\bullet\mathbf{P}_H - R^2}{\mathbf{P}_H\bullet\mathbf{P}_H - R^2 + \lambda_H}$	$\tau_2 \to \tau_F \frac{F_H}{F_{aH}}$	$\tau_{2F} = \tau_F \frac{F_H}{F_{aH}} = \frac{\mathbf{P}_F\bullet\mathbf{P}_H - R^2}{\mathbf{P}_H\bullet\mathbf{P}_H - R^2 + \lambda_H}$	
$\tau_H = \frac{\mathbf{P}_H\bullet\mathbf{G}_H + W_H}{F_H} = \frac{\mathbf{P}_H\bullet\mathbf{P}_H - R^2}{\mathbf{P}_H\bullet\mathbf{P}_H - R^2}$	$\tau_H \equiv 1$	$\tau_H = \frac{\mathbf{P}_H\bullet\mathbf{G}_H + W_H}{F_H} = \frac{\mathbf{P}_H\bullet\mathbf{P}_H - R^2}{\mathbf{P}_H\bullet\mathbf{P}_H - R^2}$	$\tau_H \equiv 1$
$\lambda_{mE} = \frac{(\mathbf{P}_{ED} - \mathbf{P}_{EB})\bullet(\mathbf{G}_{ED} - \mathbf{G}_{EB})}{4}$	$\lambda_{mE} \to \lambda_{mF}$	$\lambda_{mF} = \frac{(\mathbf{P}_{FD} - \mathbf{P}_{FB})\bullet(\mathbf{G}_{FD} - \mathbf{G}_{FB})}{4}$	
$\lambda_{mE} = -FP[\mathbf{P}_{mE}]$	$\lambda_{mE} \to \lambda_{mF}$	$\lambda_{mF} = -FP[\mathbf{P}_{mF}]$ $\lambda_{mF} \simeq \frac{1}{R^2}$ (32)	

Table 5 The parameters of the Relative Curve Measurement Theorem [5, (70)]

R = list5R ; critical R = 4.92443; τ_F approximates τ_E							
R	τ_E	τ_2	τ_F	ε_E	μ_E	F_{aH}	F_H
4.51	0.4837	0.4547	0.4829	0.0154	0.9307	4.1599	3.9099
4.92	0.9778	0.1452	0.9761	0.0004	0.9985	0.2936	0.0436
4.924	5.4304	0.0902	5.421	0.0002	0.9992	0.2542	0.0042
4.925	-3.2016	0.0737	-3.1961	0.0001	0.9994	0.2444	-0.0056
4.942	0.3812	-0.8624	0.3806	-0.0005	1.0022	0.0766	-0.1734
4.95	0.4193	42.3454	0.4185	-0.0009	1.0035	-0.0025	-0.2525

Table 6 OoA midpoint error for $\tau_E < 0$ and OoA twopoint error for $\tau_2 < 0$

R = list100R ; critical R = 99.5201; τ_F approximates τ_E

R	τ_E	τ_2	τ_F	ε_E	μ_E	F_{aH}	F_H
99.51	0.5	0.4447	0.5	0.	0.9999	2.2599	2.0099
99.92	0.5026	0.0365	0.5026	0.	1.	0.2696	0.0196
99.521	0.4997	-1.2709	0.4997	0.	1.	0.0706	-0.1794
99.525	0.5	0.6722	0.5	0.	1.	-0.7256	-0.9756
99.53	0.5	0.5726	0.5	0.	1.0001	-1.7209	-1.9709
99.54	0.5	0.5337	0.5	0.	1.0002	-3.7116	-3.9616

Table 7 OoA twopoint error for $\tau_2 < 0$

R = list400R ; critical R = 399.505; τ_F approximates τ_E							
R	τ_E	τ_2	τ_F	ε_E	μ_E	F_{aH}	F_H
399.501	0.5	0.4638	0.5	0.	1.	3.451	3.201
399.505	0.5006	0.0098	0.5006	0.	1.	0.255	0.005
399.506	0.5	0.7298	0.5	0.	1.	-0.544	-0.794
399.507	0.5	0.5931	0.5	0.	1.	-1.343	-1.593
399.508	0.5	0.5584	0.5	0.	1.	-2.1421	-2.3921
399.51	0.5	0.5334	0.5	0.	1.	-3.7401	-3.9901

Table 8 No OoA errors

1.5.2 The parameter λ_{mE} corrects the Midpoint OoA error

Paper [5, $ (70), (71), (72)] states "If we use $F_H + \lambda_{mE}$ instead of F_H then the Midpoint OoA error disappears. Unluckily the exact value of λ_{mE} for a general conic is unknown". So, λ_{mE} corrects the midpoint error and λ_{mE} is well-known for the simple circle, and λ_{mE} almost equals λ_{mF}. Hence, the decision functions of the midpoint and twopoint algorithms can be corrected as $F_H^{\text{corrected}} = F_H + \lambda_{mE} \simeq F_H + \lambda_{mF} \simeq F_H + \frac{1}{R^2}$ (Eq 41) and

$F_{aH}^{\text{corrected}} = F_{aH} - (\lambda_H - \lambda_{mE}) \simeq F_{aH} - (\lambda_H - \lambda_{mF}) \simeq F_{aH} - \lambda_H + \frac{1}{R^2}$ (Eq 42), and $F_{aH}^{\text{corrected}} \equiv F_H^{\text{corrected}}$.

So, we do not need to calculate the correction, because we can apply the simple correction $\lambda_{mE} \cong \lambda_{mF} \simeq \frac{1}{R^2}$ (32) on a curve segment with radius of curvature equal to R. Although this correction is amazingly simple, most RMD algorithms need no corrections for $R^4 >> 1$.

2. The relationships of the midpoint function F_H and twopoint function F_{aH}

(Tables 6, 7, and 8) show that there is no OoA error when F_H and F_{aH} have the same sign. As the difference between F_{aH} and F_H is positive there exists a bound $Rmin \le R \le Rmax$, corresponding with $F_H \le 0 \le F_{aH}$ and within this bound F_H and F_{aH} have a different sign. The radius Rmin is the solution of $F_H \triangleq FP[\mathbf{P}_H] == 0$ and the radius Rmax is the solution of $F_{aH} \triangleq FP[\mathbf{P}_H] + \lambda_H == 0$, hence Rmin also equals the critical radius (§ 1.2 p. 11). The sign of τ_E or τ_F determine the OoA midpoint error and the sign of τ_2 determines the OoA twopoint error. When the circle cuts the vertical line $\mathbf{P}_D\ \mathbf{P}_B$, the positive distances from (8) become $\rho_B = fdB \triangleq RPB - R$ and $\rho_D = fdD \triangleq R - RPD$, with $RPB \triangleq Norm[\mathbf{P}_B]$ and $RPD \triangleq Norm[\mathbf{P}_D]$. For an

8-connected curve, the MaxErr is smaller than equal to $\frac{\Delta^2}{2}$. The line ρ_D increases with R and the line ρ_B decreases with R, and they intersect in the point $\mathrm{Rd} \triangleq \frac{\mathrm{RPB}+\mathrm{RPD}}{2}$. For R smaller than Rd the minimum distance is $\rho_D \le 0.5$ and vice versa. The corrected midpoint decision function $F_H + \lambda_{mE} \ge 0$ selects the same minimum distance ρ_D. Therefore, Rd must also be the solution of FP[$\mathbf{P}_H$] + λ_{mE}[R]==0. But as we replace λ_{mE}[R] with $\frac{1}{R^2}$ (32), the solution $\mathrm{R}\lambda$ of $\mathrm{FP}[\mathbf{P}_H] + \frac{1}{R^2} = 0$ is slightly different from Rd and shifts $\mathbf{P}_H$ to $\mathbf{P}_{mH}$, therefore, Rmin and analogous Rmax shift to $\mathrm{R}\lambda \cong \mathrm{Rd}$. All solutions for all cases are given in (Table 9). The second part of (Table 9) approximates the square root as $(1+\frac{\mathrm{expr}}{2}) \simeq \sqrt{1+\mathrm{expr}}$ and the capital "R" becomes "r".

When we use rd to approximate OoARatioMidpoint we become 100% for all Rnd's and the approximation of OoARatioTwopoint gives 0%. Using the approximation of $\mathrm{R}\lambda$ instead of the approximation of Rd gives the expected result. So $\mathrm{r}\lambda$ must replace rd.

The approximation clearly shows that the OoARange is inversely proportional with Rnd or the rounded radii. So, the uncorrected OoA error disappears for large R's.

(Table 9) shows that for quadratic equations, the uncorrected OoA errors with small R's are mostly (more than 80%) OoA twopoint errors. This confirms the statement of [5; page 29]: "the midpoint method is better than the twopoint method (for conics)".

Formulas for Rnd	5	50	100	400
$\mathrm{Rmin}=\mathrm{Rnd}\sqrt{1-\frac{\mathrm{Rnd}-4.25}{\mathrm{Rnd}^2}}$	4.924429	49.540388	99.520098	399.505006
$\mathrm{Rd}=\mathrm{Rnd}\frac{\sqrt{1+\frac{4}{\mathrm{Rnd}^2}}+\sqrt{1+\frac{2*\mathrm{Rnd}-5}{\mathrm{Rnd}^2}}}{2}$	4.928650	49.540392	99.520099	399.505006
$\mathrm{R}\lambda=\mathrm{Rnd}(1-\frac{\mathrm{Rnd}-4.25}{2*\mathrm{Rnd}^2})*\sqrt{1+\frac{1}{\mathrm{Rnd}^4\left(1-\frac{\mathrm{Rnd}-4.25}{\mathrm{Rnd}^2}\right)^2}}$	4.929186	49.542504	99.521251	399.505313
$\mathrm{Rmax}=\mathrm{Rnd}\sqrt{1-\frac{\mathrm{Rnd}-4.5}{\mathrm{Rnd}^2}}$	4.949747	49.542911	99.521354	399.505319
OoARange = Rmax – Rmin	0.025319	0.002523	0.001256	0.000313
$\mathrm{OoARatioMidpoint}=\frac{\mathrm{Rd}-\mathrm{Rmin}}{\mathrm{OoARange}}*100$	16.673456	0.163003	0.040388	0.002506
$\mathrm{OoARatioTwopoint}=\frac{\mathrm{Rmax}-\mathrm{Rd}}{\mathrm{OoARange}}*100$	83.326544	99.836997	99.959612	99.997494
Approximation				
$\mathrm{rmin}=\mathrm{Rnd}\left(1-\frac{\mathrm{Rnd}-4.25}{2*\mathrm{Rnd}^2}\right)$	4.925	49.542500	99.521250	399.505313

~~$rd = Rnd\left(1 - \frac{Rnd - 4.5}{2 * Rnd^2}\right)$~~	~~4.95~~	~~49.545~~	~~99.522500~~	~~399.505625~~
$r\lambda = Rnd\left(1 - \frac{Rnd - 4.25}{2 * Rnd^2}\right) * \left(1 + \frac{1}{2 * Rnd^4\left(1 - \frac{Rnd - 4.25}{Rnd^2}\right)^2}\right)$	4.929187	49.542504	99.521251	399.505313
$rmax = Rnd\left(1 - \frac{Rnd - 4.25 - \lambda H}{2 * Rnd^2}\right)$	4.95	49.545	99.522500	399.505625
$OoArange = rmax - rmin \cong \frac{0.125}{Rdn}$	0.025000	0.002500	0.001250	0.000313
$OoAratioMidpoint = \frac{r\lambda - rmin}{OoArange} * 100$	16.749920	0.164502	0.040582	0.002509
$OoAratioTwopoint = \frac{rmax - r\lambda}{OoArange} * 100$	83.250080	99.835498	99.959418	99.997492

Table 9 The parameters defining the graphical relation between F_{aH} and F_H

Uncorrected RMD algorithm with Rnd ≡ 5

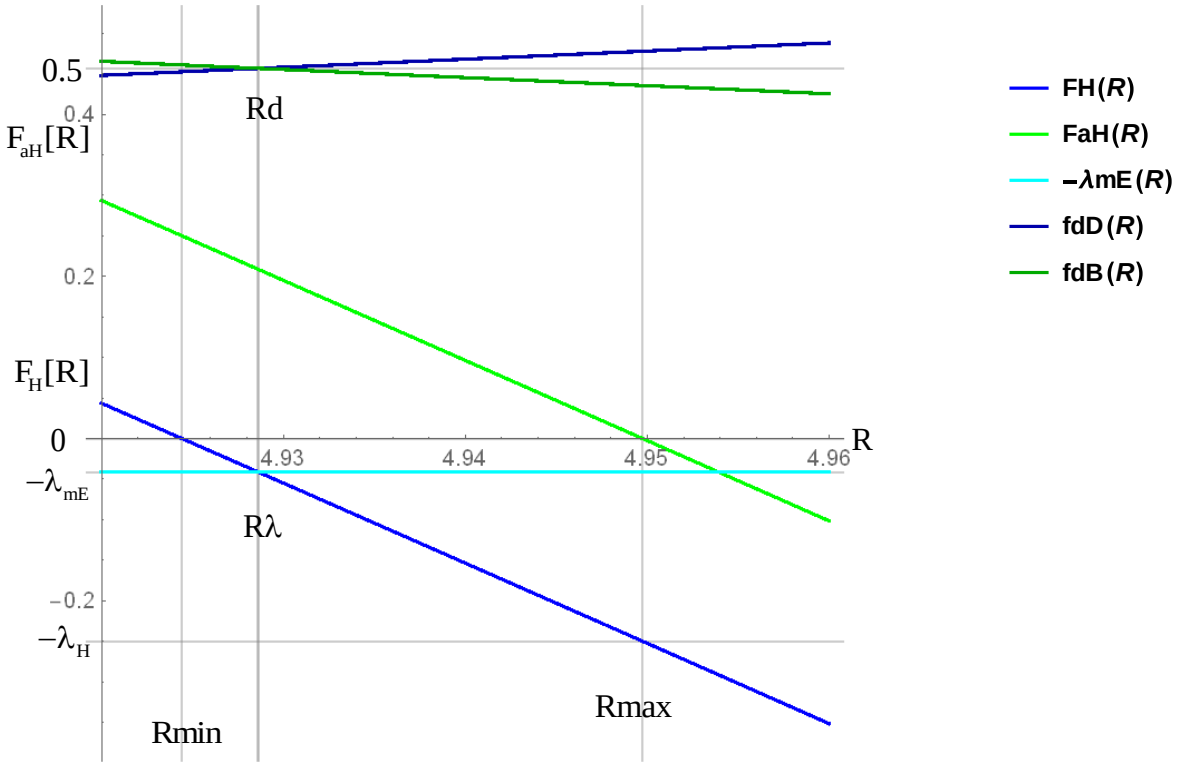


Fig 4 Uncorrected midpoint and twopoint with small R's OoA error

Circles or conics with small radii of curvature (Rnd equals 5) have midpoint- and twopoint OoA errors. But in both cases the dErr's of the OoA-points are smaller than 0.5.
The radii Rmax and Rmin determine the uncorrected OoARange.

Corrected RMD algorithm with Rnd ≡ 5

Correcting OoA does not change Rd, but both Rmin and Rmax become Rd or $R\lambda$, therefore the OoARange becomes zero.
The corrected decision functions are equal, and they intersect the zero axis for R equal to Rd or $R\lambda$.

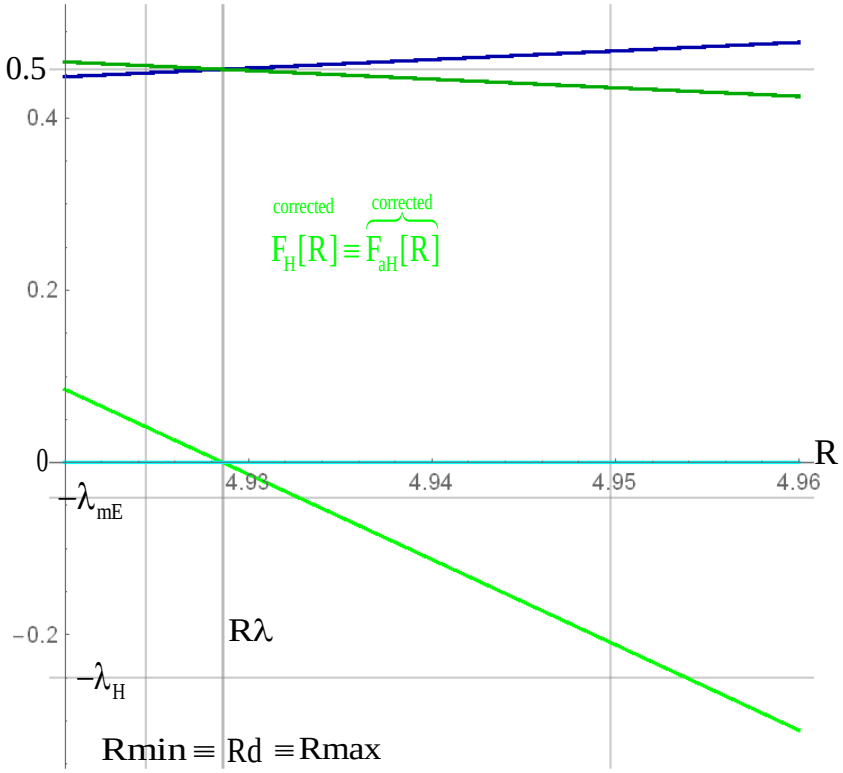


Fig. 5 The OoARange of the corrected midpoint and twopoint equals zero.
No OoA errors.

Uncorrected RMD algorithm with Rnd ≡ 400

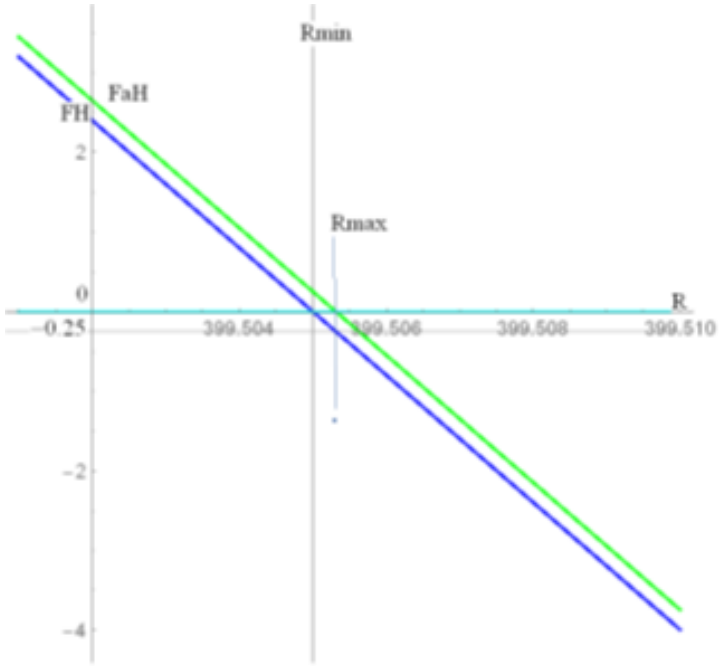


Fig. 6 For Rnd equal to 400

Circles or conics with nor small nor large radii of curvature (Rnd equal to 50 or 100) have a twopoint OoA error but no midpoint OoA error. The dErr of the OoA-points is greater than 0.5, but almost equal to 0.5.

Circles or conics with large radii of curvature (Rnd equal to four hundred or higher) have no twopoint or midpoint OoA errors. The MaxErr is smaller than 0.5.

The larger the rounded radius, the more one corrects the OoA error.

Although we detected no OoA for Rnd equal to 400, we see from (Fig. 6) that we have an OoA twopoint error for R equal to

$$\frac{\text{Rmin}+\text{Rmax}}{2}=\frac{399.505006+399.505319}{2}=399.5051625\simeq 399.5052\,.$$

The general algorithm to generate an OoA twopoint error is:

- Determine Rnd,
- Determine $\mathbf{P}_A,\mathbf{P}_B,\mathbf{P}_D,\mathbf{P}_C,\mathbf{P}_H,\mathbf{P}_V$,
- Determine the major x- or y- direction, here we assume the major x-direction,
- Determine λ_H (6A),
- Rmin $\triangleq$ Norm[$\mathbf{P}_H$],
- R(OoAtwopoint) $\triangleq$ $\text{Rmin}+\frac{\text{OoArange}}{2}$ = $\text{Rmin}+\frac{1}{2}\frac{\lambda_H}{2*\text{Rnd}}$.

Therefore, ***everyone can generate an OoA twopoint error and the last resort against that hair-splitting is OoA correcting***.

3.0 The constant feedrate algorithm for RMD curves

The 2D-polar line of a quadratic curve or the polar of an algebraic curve approximated in the region in the region $\{\mathbf{P}_{FB},\mathbf{P}_{mF},\mathbf{P}_{FD}\}$ is equivalent to

$\frac{x - x_S}{|T_{Ai}|} = \frac{y - y_S}{|T_{Aj}|} = \frac{nkey}{TAM} = \frac{npuls}{TAN} = t$, $0 \le t \le 1$ (npuls ≡ rounded arclength) with

npuls is the rounded arclength of the curve,
nkey the number of loops of the While-loop,

ATA ≙ Abs[$\mathbf{T}_A$] = { $|T_{Ai}|$, $|T_{Aj}|$ } the absolute tangent in point $\mathbf{P}_A$,

ATA ≙ Abs[Cross[$\mathbf{G}_A$]] =Abs[Cross$[GP[\mathbf{P}_A]]$ = { $|Y_A|$, $|X_A|$ } (3)

TAM ≙ Max[ATA] the maximum absolute component of { $|Y_A|$, $|X_A|$ } (Eq 43),

TAN ≙ Norm[ATA] =Norm[$\mathbf{G}_A$] (Eq 44) the norm of the tangent ATA or the gradient $\mathbf{G}_A$,

majN ≙ Switch[TAM, $|Y_A|$, 1, $|X_A|$, 2], de major x-axis corresponds with the value 1 and the major y-axis corresponds with value 2.

The PRM-cs is a pulse rate algorithm [1; 2 § 5)] that calculates $nkey = Round[\frac{TAM}{TAN} * npuls]$ or $npuls = Round[\frac{TAN}{TAM} * nkey]$ when the RMD algorithm makes a major move. Normally LMcs = TAM and LNcs = TAN and initial ACcs = $\frac{TAN}{2}$ such that $\frac{LMcs}{LNcs} = \frac{TAM}{TAN}$.

For lines LMcs and LNcs are integers, LMcs can change but LNcs has a fixed value greater or equal than LMcs.

For curves, the values of LMcs and LNcs can change. Therefore, the constant feedrate algorithm for curves becomes

$\frac{LMcs}{LNcs} = \frac{TAM}{TAN} = \frac{\frac{TAM}{TAN} * LN°}{LN°} \cong \frac{Round\,[\frac{TAM}{TAN} * LN°\,]}{LN°}$ with LN° the fixed normalized length.

The constant feedrate algorithm PRM-cs is valid for the monotonic arc segment with startpoint $\mathbf{P}_{monoS}$ and endpoint $\mathbf{P}_{monoE.}$ The initial values of the integer PRM-cs algorithm are,

LNcs ≡ LN° = $2 * Ceiling[\, Norm[\, \mathbf{P}_{monoE} - \mathbf{P}_{monoS}\,] + 0.5\,]$, (Eq 45)

While[LN° ≤ TAN, LN°=LN°+LN°].

So, LN° has a fixed and even value, such that the values of ACcs are integers,

LMcs ≙ $Floor[\frac{TAM}{TAN} * LN° + 0.5] = Round\,[\frac{TAM}{TAN} * LN°\,]$ (Eq 46),

ACcs ≙ $\frac{LN°}{2}$ (Eq 47),

The RMD algorithm updates LMcs for each point $\mathbf{P}_A$.

The PRM-cs algorithm for curves (Eq 48) becomes with initial values nkey = nuls = 0,

```
nkey = nkey + 1;
If[major axis move,
    Label[StartCS];
    Wait for interrupt Tint;
    nuls = nuls +1;
    ACcs = ACcs – LMcs;
    If[ ACcs > 0, Goto[StartCS]];
    ACcs = ACcs + LN° ;
];
```

So, the integer PRM-cs algorithm only updates nuls when the 3D-curve makes a major axis move (the 2D-curve always makes a major axis move).

The final value NPULS of npuls is the number of interrupts.
The final value nkeyM of nkey is the number of loops of the IPO.
The total arclength NPULS * Δ divided by the total time NPULS * T_{int} equals the feedrate $V = \frac{\Delta}{T_{int}}$, the interrupt frequency equals $\frac{1}{T_{int}}$ and Δ is the BLU [1; 2; § 1.1].
Every move of an 8-connected curve is always a major axis move. The final npuls NPULS measures the rounded arclength, and the value npuls[$i * T_{int}$] measures the current rounded arclength.
The rounding of LMcs (46) causes the final value of npuls to deviate from the truly rounded arclength NPULS[kcs] ≙ kcs*NPULS, the value kcs is near to one. To obtain NPULS[kcs] you must initialize LMcs with $\frac{1}{kcs} * \text{Round}[\frac{TAM}{TAN} * LN°]$.
The normal value for kcs is one or a value close to one (to correct the rounding), else the value should equal kcs = $\frac{\text{Rounded precalcuted arclength}}{\text{NPULS using kcs} \equiv 1}$ and a possible small correction to correct the rounding of (46). Avoid using the value kcs different from one, for example for a line, using kcs ≡ 1, NPULS equals Round[Norm[ATA]] and nkey equals Round[TAM]. But with kcs equal to $\frac{TAM}{Norm[ATA]}$, NPULS equals the maximum absolute coordinate Round[TAM] of the line instead of the length of the line.
The PRM-cs algorithm measures the length of the curve when kcs equals one or when kcs is near-one. The normal value for kcs is one or the value near-one (to correct the rounding) and else the value should equal kcs = $\frac{\text{Round[Precalcuted value]}}{\text{NPULS}}$ and a possible small correction to correct the rounding.
For 2D and 3D-curves, the algorithm calculates the tangent in the point $\mathbf{P}_A$, TAM, TAN and $\text{Round}[\frac{TAM}{TAN} * LN°]$ replaces LMcs.

3.1 The PRM-cs algorithm is only valid for 8-connected and 26-connected curves

The PRM-cs algorithm measures the arclength NPULS and gives constant feedrate to the midpoint- and twopoint RMD algorithms. The discrete points of the RMD algorithm are always 8-connected or 26-connected. The PRM-cs algorithm must be applied to the 8-connected points or to the major axis points of the 26-connected points.
We prove this with a counter example. The Mathematica function Table[{R Cos[θ], R Sin[θ]}, { θ , 0, 2 π, dθ }] generates the array listTC of the points $P_n = \{x_n, y_n\}$ = {R Cos[θ_n], R Sin[θ_n]} of the simple circle $x^2 + y^2 = R^2$ with $R \triangleq 50$ and $d\theta \triangleq \frac{2\pi}{160}$. The CCW tangent vector in each point P_n equals $T_n = \{-y_n, x_n\}$. The points Pn are better than the points generated with the RMD algorithm and the generated envelope with the tangents Tn is better than the envelope generated with polar lines of the RMD algorithm (§ 1.4). The result of the PRM-cs algorithm with the RMD algorithm gives the rounded length 314 and the result of the PRM-cs algorithm with the Table algorithm gives the rounded length 17919, so the PRM-cs algorithm is not valid for the Table algorithm. Sampled data systems can use the Table function to generate the sampled data points and to find the feedrate or the sampling time, one needs the length of curve.

4.0 The calculation of small offsets

The offsets of 2D-curves composed of lines and circular arcs exist for almost forty years. The tool path is an offset contour of the part contour, usually finishing uses a small offset, and pocket milling, or preformed machining use large offsets. Sometimes, some lines or circular arcs must be removed and sometimes a new line or circular arc is added (called trimming). The offset must conform with the original monotonic segments; therefore, you must add or remove lines or circular arcs. Caravantes et al [20]] extended the 2D-curves to a high degree determinantal representation for the implicit equation of the offsets to conics, quadrics and QSICs. The Pythagorean-hodograph offset curves also use high degree equations. The RMD method does not increase the degree of the equations and as the RMD uses the tangent in its constant feedrate algorithm and in its decision function, it does not really increase the complexity of the offset calculations. The offset RMD algorithm keeps simple and flexible. Of course, trimming increases the complexity, but that is how it was forty years ago.
This paper considers only small offsets and only the Mordell cubic example will be described but the 3D-offset algorithm uses the same method for the intersection of the parabolic cylinder with the one-sheet hyperboloid [20], [5; § 2.8, § 6.6 Example "e"]
Offset Conditions:

- the curve is subdivided into monotonic segments,
- the monotonic segments consist of conic or algebraic sub segments of the same type,
- the sub segments of the same type are subdivided into new segments using the extreme points,
- the small offsets consider only C^1 continuous segments (position and tangent).

The RMD-algorithms compute the best points $\mathbf{P}_A$ (or $\mathbf{P}_n$) and use the gradient $\mathbf{G}_n \equiv \mathbf{G}P[\mathbf{P}_n]$. Instead of storing $\mathbf{P}_n$ (the discrete reference curve listMordell), the offset-algorithm calculates the offset vector $\delta\mathbf{P}[\mathbf{P}_n] = \frac{\mathbf{G}P[\mathbf{P}_n]}{\text{Norm}[\mathbf{G}P[\mathbf{P}_n]]}] * \delta_{\text{Offset}}$ (49) and stores the rational $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$ in the offset path listxy. That generates the negligible (Table 11, MaxnA) offset error $\delta A[\mathbf{P}_n] \equiv \sqrt{\text{Dot}\left[\delta\mathbf{P}[\mathbf{P}_n], \delta\mathbf{P}[\mathbf{P}_n]\right]} - \text{Abs}[\delta_{\text{Offset}}]$ (50).
The offset can also change the PRM-cs algorithm as the ratio of the feedrate V_O at the offset path to the speed V_A at the part contour equals $\frac{V_O}{V_A} = \frac{R_A \pm R_O}{R_A}$. The PRM-cs of the offset curve uses for example $\mathbf{T}[\mathbf{P}_A]$ for finishing and $\mathbf{T}[\mathbf{P}_O] \equiv \mathbf{T}[\mathbf{P}_A \pm \delta\mathbf{P}[\mathbf{P}_n]]$ for roughing.
The solution of the offset problem:
Point $\mathbf{P}_n$ is an integer point and the offset point $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$ is not an integer point. Hence, the RMD algorithm cannot use the UpDown Counter [1; 2; Fig. 1], even when you use $\text{Round}[2^8(\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n])]$ (Fig. 12, c) the absolute integer increments are not {0,1} but {Δ1, Δ2} with Δ1→ 0 << 256 and Δ2→ 256.
The 32-bit Signed Binary Accumulator must replace the counter. The RMD inputs $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$ and the Accumulator reads the accumulated feedback pulses. Therefore, the Accumulator holds the non-integer servo error. The Signed Binary Accumulator has a fixed binary point (viz. nF fraction bits and nI = 31- nF integer bits) and must transfer its value to a 32-bit DAC or a 24-bit DAC. The Signed Binary Accumulator always causes rounding, and the offset error increases from $\delta A[\mathbf{P}_n]$ to $\delta A2[\mathbf{P}_n]$ (Table 11, MaxnA2) when the simulated value uses

$\mathbf{P}_n + \delta\mathbf{P}2[\mathbf{P}_n] \triangleq \frac{\text{Round}[2^8(\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n])]}{2^8}$ of (Fig. 12; d).

For all points $\mathbf{P}_n$ of the Mordell curve $\delta A[\mathbf{P}_n] \simeq 0$ and $\|\delta A2[\mathbf{P}_n]\| < 0.00204 \ll 0.5$

As the maximum offset errors are much smaller than 0.5, the MaxErr of the Offset curve equals approximately the MaxErr of the reference curve.

5. Mordell the algebraic cubic with offset and constant feedrate (22)

The residues of the updates of FB, FD, and FC are power equations in {xn, yn}. For quadratic equations, the midpoint method uses the incremental or update equation (4A), that gives for conics the update equations [21; section d; Upd XM,YM, FM]; these incremental updates avoid multiplications. Implicit algebraic equations of the third or higher degree cannot use the incremental equation (4A), but they can use forward differencing. But this section uses the brute-force for equations to present the RMD algorithm as clearly as possible. The cubic equation of the Mordell curve is $F[x,y] = x^3 - SC\, y^2 + n\, SC^3$ with $n = 8$, $\Delta = 1$, $SC = 8$ for the low-resolution Mordell and $SC = 100$ for the medium-resolution Mordell.

The RMD algorithm uses the midpoint decision function to generate

- the array listMordel of the 8-connected discrete points { $\mathbf{P}_n$ },
- the array listxy of the discrete offset points { $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$ },
- the array listxy2 of the discrete offset points { $\mathbf{P}_n + \delta\mathbf{P}2[\mathbf{P}_n]$ },
- the array listxy2C of the discrete offset points { Round[2^8 (Pn+ δP[Pn]) } ,
- the array listnA of the offset error { $\delta A[\mathbf{P}_n]$ },
- the array listnA2 of the offset error { $\delta A2[\mathbf{P}_n]$ },
- the array listErr of the error of the point $\mathbf{P}_n$ {dErr[$\mathbf{P}_n$]},
- the array listSIM of { nputs, $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$, $\mathbf{G}$P[$\mathbf{P}_n$]}.

The offset δ_{Offset} equals $0.2 * SC$.

The offset vector equals $\delta\mathbf{P}[\mathbf{P}_n] = \frac{\mathbf{G}P[\mathbf{P}_n]}{\text{Norm}[\mathbf{G}P[\mathbf{P}_n]]}] * \delta_{\text{Offset}}$.

The RMD algorithm consist of an offline part and a real-time part.

5.1 The offline part

This section writes the variables ($\mathbf{P}_n \rightarrow$ Pn) as in the $*$.nb and $*$.cdf files [22] .

- The determinations of the general gradient {XP[P], YP[P], WP[P]}
  The homogenous cubic equals $Fh[x,y,w] = x^3 - SC\, y^2\, w + n\, SC^3\, w^3$
  $X[x,y] = \frac{1}{2} * \frac{\partial F[x,y]}{\partial x}$, $Y[x,y] = \frac{1}{2} * \frac{\partial F[x,y]}{\partial y}$, and $W[x,y] = \frac{1}{2} * \frac{\partial Fh[x,y,w]}{\partial w}$ /. $w \rightarrow 1$
  The algorithm is easier to read when the point P converts the x, y-coordinates. Therefore,
  FP[P] := F[x,y] /. { $x \rightarrow$P[[1]], $y \rightarrow$P[[2]] },
  XP[P] := X[x,y] /. { $x \rightarrow$P[[1]], $y \rightarrow$P[[2]] },
  YP[P] := Y[x,y] /. { $x \rightarrow$P[[1]], $y \rightarrow$P[[2]] },
  WP[P] := W[x,y] /. { $x \rightarrow$P[[1]], $y \rightarrow$P[[2]] },
  GP[P] := {XP[P],YP[P] .

- The determination of the extreme points
  The extreme points are the intersection points of the cubic FP[P] with respectively XP[P], YP[P], and WP[P]. The function Solve of Mathematica gives the extreme points, using
  Solve[{ FP[P]==0, XP[P]==0 }, P, reals],
  Solve[{ FP[P]==0, YP[P]==0 }, P, reals],
  Solve[{ FP[P]==0, WP[P]==0 }, P, reals].

  With PS = {xS, yS} ≙ { -2* SC, 0 }, PE = {xE, yE} ≙ { $SC*2*2^{\frac{1}{3}}$, $SC*2*\sqrt{6}$ }, and Ph = {xh, yh} ≙ { 0, $SC*2*\sqrt{2}$ },
  the respectively extreme points become
  listPX = {{ Ph[[1]], -Ph[[2]] }, {{ Ph[[1]], Ph[[2]] }},
  listPY = {{ PS[[1]]], PS[[2]] }},
  listPW = {{ PE[[1]], -PE[[2]] }, {{ PE[[1]], PE[[2]] }}.
  The upper part moves from PS to Ph to PE. The extreme points subdivide the path into monotonic paths with monotonic direction {Sx,Sy}. In this case the monotonic direction from PS to Ph is {1,1} and the monotonic direction from Ph to PE is also {1,1}, the monotonic segment PS to PE is not subdivided. The while-loop of the real-time algorithm uses integers therefore, the RMD algorithm rounds all non-integer points before starting the while-loop. The startpoint PS becomes PS = {-16, 0} and the endpoint PE becomes PE = {20, 39}, the monotonic direction equals {Sx, Sy} = {1, 1} = Sign[PE-PS].
- The offline calculation of the arclength
  The constant feedrate algorithm PRM-cs, also calculates the current rounded arclength npuls and the final NPULS should be close to the rounded offline calculated arclength.
  The offline calculated arclength uses the ImplicitRegion $\Re$ of Mathematica and the truly projections of the path, viz. $\Re$ = ImplicitRegion[FP[P] == 0 && PS[[1]] ≤ x ≤ PE[[1]] && PS[[2]] ≤ y ≤ PE[[2]], {x,y}].
  The calculated arclength =Quiet[N[ArcLength[$\Re$, Method → "Integrate"]]].
  The calculated arclength is the arclength of the upper part of the Mordell curve starting in point **P**S, going to point **P**h and ending in point **P**E (Table 10).

Mathematica gives		RMD algorithm gives	
Arclength from PS to PE		Arclength from PS_{int} to PE_{int}	
For SC equal to 8	For SC equal to 100	For SC equal to 8	For SC equal to 100
59.7535	746.919	NPULS = 60	NPULS = 747

Table 10 The arclength of upper part of the Mordell curve from PS to PE

The absolute minimum distance of a point Pn={xn, yn} to the curve FP[P] equals dErr[Pn]:=N[RegionDistance[$\Re$, Pn]. One generates listErr in the While-loop and one calculates MaxErr = Max[listErr] when the While-loop has finished.

5.2 The real-time part

5.2.1 The initializations of the RMD algorithm

- The startpoint PS, the endpoint PE,
  The startpoint is PS = {xS, yS} = Round[PS],
  The endpoint is PE = {xE, yE} = Round[PE],
  The monotonic direction {Sx,Sy}=Sign[PE-PS],
  The best selected candidate point becomes Pn = {xn, yn} = PS,
- The reference point PA = {xA, yA} = Pn (Fig. 3)
- The initializations of the storing arrays
  listxy=listyxy2=listxy2C=listMordel=listnA=listnA2=listErr={},

- The initializations of the arclength and the number of loops
  npuls=nkey=0
- The initializations of the variables of the constant feedrate algorithm
  LNcs = LN° = 2* Ceiling[Norm[PE-PS]+0.5] is the normalized length,
  ACcs = $\frac{\text{LNcs}}{2}$,
- The initialization and update of the PRM-cs algorithm (constant feedrate algorithm)
  ATA=Abs[GP[PA]],
  TAM=Max[ATA],
  TAN=N[Norm[ATA]],
  LMcs=Round[$\frac{\text{TAM}}{\text{TAN}} * \text{LNcs}$].
- Computing the candidate points
  PB = {xB, yB} = PA + {Sx, 0 },
  PD = {xD, yD} = PA + {Sx, Sy},
  PC = {xC, yC} = PA + {0 , Sy},
- The preparation of the Offsets and the offset errors, define

  $\delta P[Pn\_] := \frac{GP[Pn]}{Norm[GP[Pn]]} * \delta Offset$ (Eq 49),

  $\delta A[Pn\_] := \sqrt{Dot[\delta P[Pn], \delta P[Pn]]} - Abs[\delta Offset]$ (Eq 50),

  $\delta P2C[Pn\_] := Round[2^8 * \delta P[Pn]]$ (Eq 51),

  $\delta A2C[Pn\_] := \sqrt{Dot[\delta P2C[Pn], \delta P2C[Pn]]} - 2^8 * Abs[\delta Offset]$ (Eq 52),

  $\delta P2[Pn\_] := \frac{\delta P2C[Pn]}{2^8}$ (Eq 53),

  $\delta A2[Pn\_] := \frac{\delta A2C[Pn]}{2^8}$ (Eq 54),

- The initialization of the arrays
  listxy = Join[listxy, { Pn+ δP[Pn] }],
  listxy2 = Join[listxy, { Pn+ δP2[Pn] }],
  listxy2C = Join[listxy, { Round[2^8 (Pn+ δP[Pn]) }],
  listMordell = Join[listMordell, { Pn }],
  listnA = Join[listnA, { δA[Pn] }],
  listnA2 = Join[listnA2, { δA2[Pn] }],
  listErr = Join[listErr, { dErr[Pn] }],
  listSIM = Join[listSIM, {{ npuls, $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$, **G**P[$\mathbf{P}_n$] }}].

5.2.2 The While-loop RMD algorithm

Simplification of the decision functions (40)

If FaM $\triangleq$ $\frac{FP[PB] + FP[PC]}{2}$ > 0 the moves are in the major y-direction, else the moves are in the major x-direction. The decision function in the major y-direction is *move from PA If[(FD-FC)*FaV > 0, to PC, to PD]* and the decision function in the major x-direction is *move from PA If[(FB-FD)*FaH < 0, to PB, to PD]*.

As the residues of the inside of the Mordell curve are positive, and the differences FB -FD and FD – FC are positive for the monotonic direction {1, 1}, the decision functions become simpler, as Sign[FB - FD] ≡ Sign[FD – FC] ≡ one.

The End-of-While Filter

The While-loop stops when the current point Pn equals the end point PE. Regularly, the Pn misses PE. That is quite understandable, because two 8-bit connected lines have often no intersection (4-connected curves do not have that problem). The Filter avoids that problem. The initialization {δx, δy} = {1, 1} enables the update equations (see update of the candidate points). Normally the x- and y-updates of the candidate points occurs at the end of the While-loop. If (xn == xE), the the value of xn does not changes anymore (δx= 0) and If[yn == yE), the value of yn does not changes anymore (δy = 0).

```
{δx, δy} = {1, 1}  (* enable the x and y increments *)
While[Pn ≠ PE,
    nkey = nkey + 1 ; (* increment the loop number *)
    (* PRM algorithm  ≡ constant feedrate algorithm ≡ arclength algorithm *)
      Label[ StartCS ] ;
      npuls = npuls + 1 ;
      ACcs = ACcs – LMcs;
      If[ ACcs > 0, Goto[StartCS];
      ACcs = ACcs + LNcs;
    (* Computing the residues of the Mordell curve at the candidate points *)
    FB = FP[PB] ;
    FC = FP[PC] ;
    FD = FP[PD] ;
    FaM = (FB + FC)/2 ;
    If[ FaM > 0,  (* major y-direction and FD – FC > 0 *)
            FaV = (FD + FC)/2 ;
            If[ FaV > 0, Pn = {xn, yn} = PC , Pn = {xn, yn} = PD ] ;
    ,       (* major x-direction and FB – FD > 0 *)
            FaH = (FB + FD)/2 ;
            If[ FaH < 0, Pn = {xn, yn} = PB , Pn = {xn, yn} = PD ] ;
    ] ;
    (* fill in the arrays *)
    listxy = Join[ listxy, { Pn+ δP[Pn] } ] ;
    listxy2 = Join[ listxy2, { Pn+ δP2[Pn] } ];
    listxy2C = Join[ listxy2C, { Round[ 2^8 (Pn+ δP[Pn]) } ];
    listMordell = Join[ listMordell, { Pn } ] ;
    listnA = Join[ listnA, { δA[Pn] } ] ;
    listnA2 = Join[ listnA2, { δA2[Pn] } ];
    listErr = Join[ listErr, { dErr[Pn] } ] ;
    listSIM = Join[ listSIM, {{ nputs, Pn + δP[Pn], GP[ Pn ] }} ] ;
    (* update the reference point PA *)
    PA = {xA, yA} = Pn ;
    (* update the constant feedrate algorithm *)
    ATA=Abs[GP[PA]] ;
    TAM=Max[ATA] ;
    TAN=N[Norm[ATA]] ;
    LMcs=Round[ (TAM/TAN) * LNcs ];
```

```
    (* Apply the End-of-While FILTER *)
    If[xn == xE, δx = 0];
    If[yn == yE, δy = 0];
    (* update the candidate points *)
    PB = {xB, yB} = PA + {δx*Sx, 0 } ;
    PD = {xD, yD} = PA + {δx*Sx, δy*Sy} ;
    PC = {xC, yC} = PA + {0 , δy*Sy} ;
]  (* End of While *)
NPULS = npuls ;
nkeyM = nkey ;
```

$\text{listSIM}_{\text{offset}}$ = listSIM. (55)

5.3 The results of the RMD twopoint algorithm

(Table 11) gives the rounded arclength NPULS, MaxErr = Max[listErr], MaxnA = Max[listnA] for respectively the Mordell curve with SC equal to 8 and 100.

	SC = 8	SC = 100	Info
NPULS	60	747	Round arclength
MaxErr	0.466668	0.494178	Max[listErr]
MaxnA	$4.44089*10^{-16}$	$3.55271*10^{-15}$	Max[listnA]
MaxnA2	0.0020388	0.00257742	Max[listnA2]
nkeyM	54	674	Number of Loops

Table 11 The Offset error has no practical influence on the MaxErr

The MaxErr of the Offset curve is also smaller than 0.5 and equals the MaxErr of the reference curve.

$$\mathbf{r}_{\text{offset}} = \text{listSIM}_{\text{offset}} = \{\text{npuls}[i*T_c], \mathbf{P}[i*T_c], \mathbf{GP}[i*T_c]\}\Big|_{i=1}^{i=\text{nkeyM}} \quad \text{(Eq 55).}$$

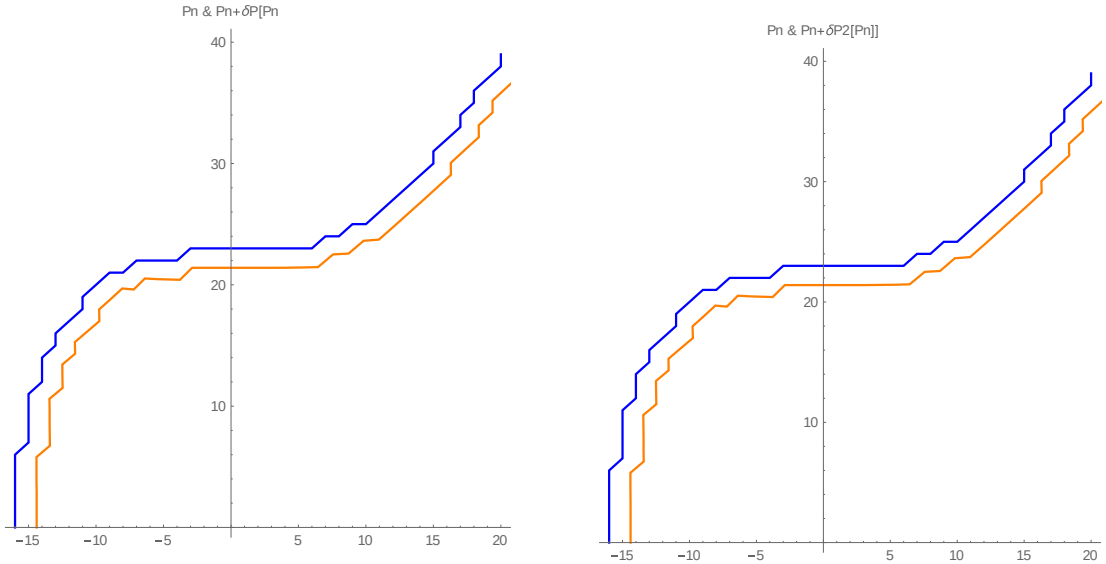


Fig. 7 The integer reference curve Pn and the non-integer offset curve Pn+δP[Pn].

Offset curve Pn + δP2[Pn] $\equiv \frac{\text{Round}[256(\text{Pn} + \delta\text{P}[\text{Pn}])]}{256}$ and integer reference curve Pn.

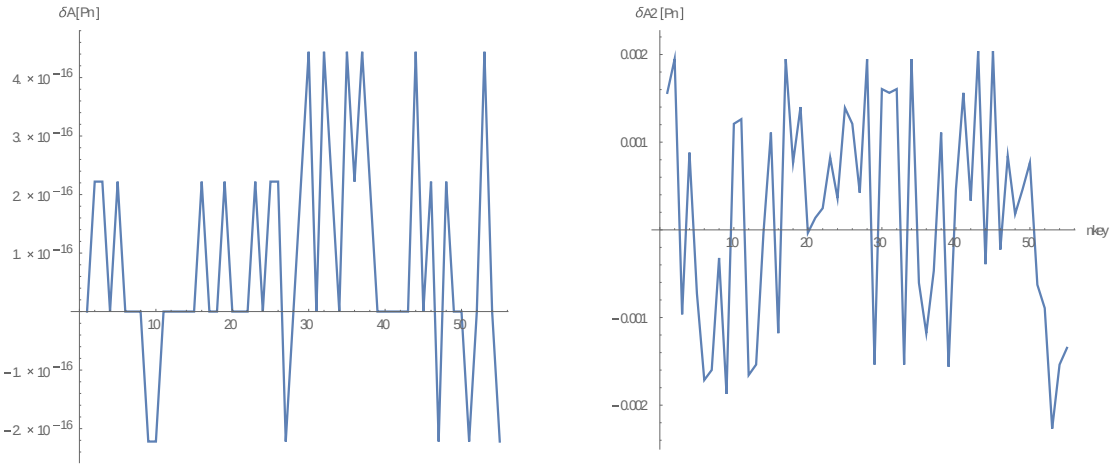


Fig.9 The Offset error δA is smaller than 5.10^{-16}. The Offset Error $\delta A2$ is smaller than 0.00204

As the maximum offset error is much smaller than 0.5, the MaxErr of the Offset curve equals approximately the MaxErr of the reference curve.

$\mathbf{P}_n$		$\mathbf{P}_n+\delta\mathbf{P}[\mathbf{P}_n]$		$\text{Round}[2^8(\mathbf{P}_n+\delta\mathbf{P}[\mathbf{P}_n])]$		$\mathbf{P}_n+\delta\mathbf{P}2[\mathbf{P}_n]$	
-16	0	-14.4	0.	-3686	0	-14.3984	0
-16	1	-14.4003	0.966674	-3686	247	-14.3984	0.964844
-16	2	-14.4014	1.93339	-3687	495	-14.4023	1.93359
-16	3	-14.4031	2.90019	-3687	742	-14.4023	2.89844
-16	4	-14.4055	3.86713	-3688	990	-14.4063	3.86719
-16	5	-14.4086	4.83423	-3689	1238	-14.4102	4.83594
-16	6	-14.4124	5.80154	-3690	1485	-14.4141	5.80078
-15	7	-13.4216	6.7381	-3436	1725	-13.4219	6.73828
-15	8	-13.428	7.7019	-3438	1972	-13.4297	7.70313
-15	9	-13.4352	8.66618	-3439	2219	-13.4336	8.66797
-15	10	-13.4431	9.63097	-3441	2466	-13.4414	9.63281
-15	11	-13.4518	10.5963	-3444	2713	-13.4531	10.5977
-14	12	-12.479	11.5034	-3195	2945	-12.4805	11.5039
-14	13	-12.4916	12.4664	-3198	3191	-12.4922	12.4648
-14	14	-12.5048	13.4304	-3201	3438	-12.5039	13.4297
-13	15	-11.5538	14.3154	-2958	3665	-11.5547	14.3164
-13	16	-11.5717	15.2788	-2962	3911	-11.5703	15.2773
-12	17	-10.646	16.1475	-2725	4134	-10.6445	16.1484
-11	18	-9.74658	1755	-2495	4353	-9.74609	1739
-11	19	-9.77334	17.9727	-2502	4601	-9.77344	17.9727
-10	20	-8.90569	18.8327	-2280	4821	-8.90625	18.8320
-9	21	-8.06237	19.7035	-2064	5044	-8.06250	19.7031
-8	21	-7.20618	19.6108	-1845	5020	-7.20703	19.6094
-7	22	-6.38342	20.5236	-1634	5254	-6.38281	20.5234
-6	22	-5.53068	20.4704	-1416	5240	-5.53125	20.4688
-5	22	-4.66658	20.4351	-1195	5231	-4.66797	20.4336
-4	22	-3.78382	20.4147	-969	5226	-3.78516	20.4141
-3	23	-2.88292	21.4043	-738	5479	-2.88281	21.4023
-2	23	-1.94785	21.4008	-499	5479	-1.94922	21.4023
-1	23	-0.986957	21.4001	-253	5478	-0.988281	21.3984
0	23	0.	21.4	0	5478	0	21.3984
1	23	1.01304	21.4001	259	5478	1.01172	21.3984
2	23	2.05215	21.4008	525	5479	2.05078	21.4023
3	23	3.11708	21.4043	798	5479	3.11719	21.4023
4	23	4.20694	21.4134	1077	5482	4.20703	21.4141
5	23	5.31952	21.4322	1362	5487	5.32031	21.4336

6	23	6.45056	21.4647	1651	5495	6.44922	21.4648
7	24	7.57202	22.5057	1938	5761	7.57031	22.5039
8	24	8.71554	22.5689	2231	5778	8.71484	22.5703
9	25	9.83072	23.6326	2517	6050	9.83203	23.6328
10	25	10.96	23.72	2806	6072	10.9609	23.7188
11	26	12.052	24.7944	3085	6347	12.0508	24.7930
12	27	13.1314	25.8686	3362	6622	13.1328	25.8672
13	28	14.199	26.9405	3635	6897	14.1992	26.9414
14	29	15.256	2888	3906	7170	15.2578	2878
15	30	16.3039	29.0728	4174	7443	16.3047	29.0742
15	31	16.2893	30.0526	4170	7693	16.2891	30.0508
16	32	17.3313	31.1125	4437	7965	17.3320	31.1133
17	33	18.3665	32.1678	4702	8235	18.3672	32.1680
17	34	18.3553	33.1496	4699	8486	18.3555	33.1484
18	35	19.3864	34.2013	4963	8756	19.3867	34.2031
18	36	19.3765	35.1843	4960	9007	19.3750	35.1836
19	37	20.4039	36.2326	5223	9276	20.4023	36.2344
20	38	21.4273	37.2769	5485	9543	21.4258	37.2773
20	39	21.4195	38.2618	5483	9796	21.4180	38.2617

Table 12 a. The integer 8-connected upper Mordell curve $\mathbf{P}_n$

b. The non-integer offset curve $\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n]$,

c. The integer offset curve $\mathrm{Round}[2^8(\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n])]$,

d. The simulated Offset curve $\mathbf{P}_n + \delta\mathbf{P}2[\mathbf{P}_n] = \dfrac{\mathrm{Round}[2^8(\mathbf{P}_n + \delta\mathbf{P}[\mathbf{P}_n])]}{2^8}$

The $\mathrm{listSIM} = \{\mathrm{npuls}[i * T_c], \mathbf{P}[i * T_c], \mathbf{GP}[i * T_c]\}\Big|_{i=1}^{i=\mathrm{nkeyM}}$ array

The CAD system can integrate the RMD-algorithms when the RMD-algorithm generates the array to rigid simplified CNC machine tools, which convert these arrays to real-time constant feedrate listSIM-algorithms.
The listSIM-algorithms can be used as the benchmark of all real-time curve algorithms.
This paper shows that

- you can correct as well the midpoint as the twopoint method,
- both methods have no OoA when using the corrected decision variables,
- both methods can be used without correcting the decision variables when Rnd >> 1.

The machining demands that the feedrate is constant and that the path planning can compensate the tool radius; that means that the calculation of the offset curve of the discrete connected curve must be easy, flexible, fast and the maximum error must be bounded and preferably minimal.

6. Conclusion

The RMD algorithm consists of the PRM-cs algorithm, and the incremental step algorithm based on the RMD method. The incremental steps must generate 8-connected 2D-curves or 26-connected 3D-curves. The PRM-cs algorithm gives constant feedrate to the RMD algorithm and measures the rounded arclength of the incremental steps. The PRM-cs algorithm integrates only in algorithms that generate 8- or 26-connected curves.
The maximum error MaxErr must be bounded and minimal. The MaxErr of an 8-connected curve is smaller than $\frac{1}{2}\Delta$ and the MaxErr of a 26-connected curve is smaller than $\frac{\sqrt{2}}{2}\Delta$.
The right triangle altitude theorem states that the the correction parameter is the square of the altitude of the right triangle $\{\mathbf{P}_0, \mathbf{P}_E, \mathbf{P}_{EB}\}$ and almost the square of the altitude of the right triangle $\{\mathbf{P}_0, \mathbf{P}_F, \mathbf{P}_{FB}\}$.

The correction parameter $\lambda_{mE} \cong \lambda_{mE} \simeq \frac{1}{R^2}$ for $R^4 \gg 1$.

The curve is the envelope of the family of tangent lines, the quadratic curve is almost the envelope of the polar lines of the midpoint RMD algorithm and the algebraic curve is almost the envelope of the approximate polar lines of the twopoint RMD algorithm. Therefore, the difference of the squared distances of the points $\mathbf{P}_u$ and $\mathbf{P}_v$ to respectively the quadratic curve and the algebraic curve becomes,

$$\mathrm{Sign}\left[\overbrace{\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v]}^{\text{curve}}\right] \equiv \mathrm{Sign}\left[d_M[\mathbf{P}_u,\mathbf{P}_v]\right] \equiv \mathrm{Sign}[\tau_E] * \overbrace{\mathrm{Sign}[(FP[\mathbf{P}_u]-FP[\mathbf{P}_v]) * F_H]]}^{\text{midpoint criterium}},$$

$$\mathrm{Sign}\left[\overbrace{\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v]}^{\text{curve}}\right] \equiv \mathrm{Sign}\left[d_2[\mathbf{P}_u,\mathbf{P}_v]\right] \equiv \mathrm{Sign}[\tau_2] * \overbrace{\mathrm{Sign}[(FP[\mathbf{P}_u]-FP[\mathbf{P}_v]) * F_{aH}]]}^{\text{two-point criterium}}.$$

This distance measurement is OoA when τ_E or τ_2 is negative and both parameters cannot be negative. Adding $\frac{1}{R^2}$ to F_H corrects the midpoint criterium.

The OoARange is inverse proportional with the rounded radius of the curvature, hence, the uncorrected OoA error disappears for large R's. Although the corrections for quadratic curves are amazingly simple, one rarely applies the corrections, because the OoARange can be neglected. In practice, the midpoint and twopoint RMD algorithms apply to 2D- and 3D-curves without correction of the decision functions, except for small radii.
Adding constant feedrate and rounded arclength in real time is extremely easy.
The machining demands that the feedrate is constant and that the path planning can compensate the tool radius; that means that the calculation of the offset curve of the discrete connected curve must be easy, flexible, fast and the maximum error must be bounded and preferably minimal. The small tool offset RMD algorithm uses the same algorithm as the non-offset RMD algorithm and the MaxErr of the discrete offset points equals the MaxErr of the discrete connected points.
The listSIM-algorithm can be used as the benchmark of all real-time curve algorithms and listSIM-algorithms integrate into rigid simplified CNC machine tools.

7. Appendix

A1.1 General conic equation is an implicit equations of the second degree:

$$F_P = FP[\mathbf{P}] = \mathbf{P}\bullet\mathbf{G}P[\mathbf{P}] + WP[\mathbf{P}] = F[x,y] = \begin{pmatrix} x & y & 1 \end{pmatrix} \begin{pmatrix} A_1 & D_1 & I_1 \\ D_1 & B_1 & J_1 \\ I_1 & J_1 & M_1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \text{..... (Eq 1A)}$$

A1.2 Residue of the general conic equation in the point $\mathbf{P}_H \triangleq \frac{\mathbf{P}_B + \mathbf{P}_D}{2}$ equals

$$F_H = FP[\mathbf{P}_H] = \mathbf{P}_H\bullet\mathbf{G}P[\mathbf{P}_H] + WP[\mathbf{P}_H] \quad \text{..... (Eq 2A)}$$

A1.3 La Hire's theorem or the switching property:
If a point $\mathbf{P}_B$ lies on the polar (line) of $\mathbf{P}_D$, then $\mathbf{P}_D$ lies on the polar (line) on $\mathbf{P}_B$

$$\mathbf{P}_B\bullet\mathbf{G}P[\mathbf{P}_D] + WP[\mathbf{P}_D] \equiv \mathbf{P}_D\bullet\mathbf{G}P[\mathbf{P}_B] + WP[\mathbf{P}_B] \quad \text{..... (Eq 3A)}$$

Proof:

The transpose of the symmetrical matrix $\begin{pmatrix} x_B & y_B & 1 \end{pmatrix}\begin{pmatrix} A_1 & D_1 & I_1 \\ D_1 & B_1 & J_1 \\ I_1 & J_1 & M_1 \end{pmatrix}\begin{pmatrix} x_D \\ y_D \\ 1 \end{pmatrix}$ equals the matrix itself.

A1.4 Incremental equation :

$$F_D - F_B = FP[\mathbf{P}_D] - FP[\mathbf{P}_B] \equiv 2*(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}P[\frac{\mathbf{P}_D + \mathbf{P}_B}{2}] \equiv 2*(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}P[\mathbf{P}_H] \quad \text{..... (Eq 4A)}$$

Conics often apply (4A) such that $\mathrm{Sign}\left[(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}P[\mathbf{P}_H]\right] \equiv \mathrm{Sign}\left[FP[\mathbf{P}_D] - FP[\mathbf{P}_B]\right]$.

The incremental equation is also used as the update equation of the quadratic equations.
The update equations of implicit algebraic equations is more involved.
Proof of (4A): (apply (2A) and (3A)

Apply (2A) for respectively $\mathbf{P}_H \mapsto \{\mathbf{P}_D, \mathbf{P}_B\}$.

Make the difference $F_D - F_B$ and replace $W[\mathbf{P}_D] - W[\mathbf{P}_B]$ with $\mathbf{P}_D\bullet\mathbf{G}P[\mathbf{P}_B] - \mathbf{P}_B\bullet\mathbf{G}P[\mathbf{P}_D]$.

As $\mathbf{G}P[\mathbf{P}]$ is linear in $\mathbf{P}$, apply $\mathbf{G}P[\mathbf{P}_D] + \mathbf{G}P[\mathbf{P}_B] \equiv \mathbf{G}P[\mathbf{P}_D + \mathbf{P}_B] = 2*\mathbf{G}P[\mathbf{P}_H]$

A1.5 Arithmetic mean equation for quadratic and algebraic equations of the 3rd degree

A1.5.1 The arithmetic mean F_{aH} relates the twopoint and midpoint method.

$$F_{aH} \triangleq \frac{F_B + F_D}{2} \equiv F_H + \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet(\mathbf{G}P[\mathbf{P}_D] - \mathbf{G}P[\mathbf{P}_B])}{4} \quad \text{..... (Eq 5A)}$$

$$\lambda_H = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet(\mathbf{G}P[\mathbf{P}_D] - \mathbf{G}P[\mathbf{P}_B])}{4} \quad \text{..... (Eq 6A)}$$

The arithmetic mean F_{aH} is the main decision variable of the twopoint algorithm.

Proof: for conics (can be extended to quadrics)

Apply the incremental equation respectively on $\frac{F_B - F_H}{2}$ and $\frac{F_D - F_H}{2}$, add the equations,

applying $\mathbf{P}_B - \mathbf{P}_H \equiv \frac{\mathbf{P}_B - \mathbf{P}_D}{2}$, $\mathbf{P}_D - \mathbf{P}_H \equiv -\frac{\mathbf{P}_B - \mathbf{P}_D}{2}$, $\mathbf{G}P[\frac{\mathbf{P}_B + \mathbf{P}_H}{2}] \equiv \mathbf{G}P[\mathbf{P}_B + \frac{\mathbf{P}_D}{2}]$,

$\mathbf{G}P[\frac{\mathbf{P}_D + \mathbf{P}_H}{2}] \equiv \mathbf{G}P[\mathbf{P}_D + \frac{\mathbf{P}_B}{2}]$ gives

$$F_{aH} - F_H = \frac{\mathbf{P}_D - \mathbf{P}_B}{2}\bullet(\mathbf{G}P[\mathbf{P}_D + \frac{\mathbf{P}_D}{2}] - \mathbf{G}P[\mathbf{P}_B + \frac{\mathbf{P}_B}{2}]) = \frac{\mathbf{P}_D - \mathbf{P}_B}{2}\bullet\mathbf{G}P[\frac{\mathbf{P}_D - \mathbf{P}_B}{2}] = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}P[\mathbf{P}_D - \mathbf{P}_B]}{4} .$$

For circles with $\mathbf{G}P[\mathbf{P}_B] \equiv \mathbf{P}_B = \{2, \mathrm{Rnd}\}$ and $\mathbf{G}P[\mathbf{P}_D] \equiv \mathbf{P}_D = \{2, \mathrm{Rnd} + \mathrm{Sy}*\Delta\}$:

$$\lambda_H = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet(\mathbf{P}_D - \mathbf{P}_B)}{4} = \frac{\Delta^2}{4} \quad \text{..... (Eq 7A)}$$

A1.5.2 For an algebraic equation of the 3rd degree, but not for an algebraic equation of the 4th or higher degree, $F_{aH} - F_H = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{G}P[\mathbf{P}_D - \mathbf{P}_B]}{4}$.

Proof:

Put $F[x,y] = A1*x^3 + A2*x^2*y + A3*x*y^2 + A4*y^3$ and

$$\mathbf{GP}[\mathbf{P}] \equiv \mathbf{G}[x,y] \triangleq \frac{1}{2}\text{Grad}[F[x,y],\{x,y\}] = \frac{1}{2}\left\{\frac{\partial F[x,y]}{\partial x}, \frac{\partial F[x,y]}{\partial y}\right\}.$$

Hence, $\mathbf{GP}[\mathbf{P}] \equiv \mathbf{G}[x,y] = \frac{1}{2}\{3\,A1\,x^2 + 2\,A2\,x\,y + A3\,y^2,\ A2\,x^2 + 2\,A3\,x\,y + 3\,A4\,y^2\}$.

For $F_B = F[x_B, y_B]$, $F_D = F[x_D, y_D]$, $F_H = F[x_H, y_H] = F\left[\frac{x_B + x_D}{2}, \frac{y_B + y_D}{2}\right]$,

$F_{aH} \triangleq \frac{F_B + F_D}{2}$, the difference becomes

$$F_{aH} - F_H = \frac{(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{GP}[\mathbf{P}_D - \mathbf{P}_B]}{4} = \frac{A3}{2} + \frac{3}{8}A4(-1 + 2\,\text{Rnd})$$

A1.5.3 Squared Distance of the point $\mathbf{P}_B$ to the Line $\mathbf{P}\bullet\mathbf{GP}[\mathbf{P}_H] + \mathbf{WP}[\mathbf{P}_H]$ equals

$$\boldsymbol{\rho}_L^2[\mathbf{P}_B] = \frac{(\mathbf{P}_B\bullet\mathbf{GP}[\mathbf{P}_H] + \mathbf{WP}[\mathbf{P}_H])^2}{(\mathbf{GP}[\mathbf{P}_H])^2} \quad \text{..... (Eq 8A)}$$

A1.5.4 Difference of the squared distances of the points $\mathbf{P}_B$ and $\mathbf{P}_D$ to the Line $\mathbf{P}\bullet\mathbf{GP}[\mathbf{P}_H] + \mathbf{WP}[\mathbf{P}_H]$ is the polar line with respect to the midpoint $\mathbf{P}_H$ of $\mathbf{P}_B$ and $\mathbf{P}_D$:

$$\boldsymbol{\rho}_L^2[\mathbf{P}_D] - \boldsymbol{\rho}_L^2[\mathbf{P}_B] = \frac{2*(\mathbf{P}_D - \mathbf{P}_B)\bullet\mathbf{GP}[\mathbf{P}_H]*(\mathbf{P}_H\bullet\mathbf{GP}[\mathbf{P}_H] + \mathbf{WP}[\mathbf{P}_H])}{(\mathbf{GP}[\mathbf{P}_H])^2} \quad \text{..... (Eq 9A)}$$

$$\boldsymbol{\rho}_L^2[\mathbf{P}_D] - \boldsymbol{\rho}_L^2[\mathbf{P}_B] = \frac{(F_D - F_B)*F_H}{(\mathbf{GP}[\mathbf{P}_H])^2} \quad \text{..... (Eq 10A)}$$

Proof:
Applying (8A) gives (9A)
Applying (9A), (2A) and (4A) gives (10A)

8. References

1 Huypens V., "Constant Speed Lines – Curves – NURBS Reference Pulse IPOs (part I)," SPRINGER NATURE, Int. J Adv Manuf Technol 111, 1247-1275 (2020). https://doi.org/10.1007/s00170-020-05339-1.

2 Huypens V., "Constant Speed Lines – Curves – NURBS Reference Pulse IPOs (part I] ", Font 12 pt. and one column, https://www.researchgate.net/profile/Valere-Huypens/publications

3 Huypens V., "Perfect 3D-Curve RMDPL-IPOs & Bresenham's 3D-Curve Algorithm (Part 2 & 3) Version 1, July 2021, DOI 10.13140/RG 2.2.26648.11521, DOI 2.2.2664810.13140/RG .11521,https://www.researchgate.net/profile/Valere-Huypens/publications

4 Huypens V., "Relative Minimum Distance 3D-Curve and 3D-Offsets Real -Time Constant Feedrate Algorithms, DOI: 10.13140/RG.2.2.10011.95522, Year 2022, https://www.researchgate.net/profile/Valere-Huypens/publications

5 Huypens V., "Relative Squared Distances to a Conic," International Journal of Computer Graphics & Animation (IJCGA), 5, No, 1, pp. 17-37, January 2015.

6 V. Huypens, "Chapter 3: The Sign Corrected Midpoint Decision Variable Selects the Candidate Point with the Minimum Euclidean Distance to the Conic," 2008 3rd International Conference on Geometric Modeling and Imaging, 2008, pp. 15-20, doi: 10.1109/GMAI.2008.8.

7 Georges Salmon, "A Treatise on Conic Sections," 5th Edition, 1869, London: Longmans, Green, Reader and Dyer.

8 Donald L. Vossler, "Exploring Analytic Geometry with Mathematica," Academic Press, 1999, pages 865. The book is out of print. The PDF edition is available and the accompanying CD, including the analytic geometry software system called Descarta2D, with 30 Mathematica notebooks.

9 Taubin Gabriel, "Distance approximations for rasterizing implicit curves," ACM Transactions on Graphics, Volume 13, Issue 1, Jan. 1994, DOI:10.1145/174462.174531

10 Bresenham J.E., "Algorithm for computer Control of a digital plotter, "IBM Systems Journal, Vol. 4, pp. 25-30, 1965.

11 Bresenham J.E., "A Linear Algorithm for Incremental Digital Display of Circular Arcs, "Communications of the ACM, Vol. 20, No. 2, pp. 100-106, Feb. 1977.

12 Bresenham J.E. et. al., "Fundamental Algorithms for Computer Graphics," NATO ASI Series, Series F: Computer and Systems Sciences, Vol. 17, 1985, Springer-Verlag.

13 Pitteway M.L.V., "Algorithm for Drawing Ellipses or Hyperbolae with a Digital Plotter," Computer J., 10(3), November 1967, 282-289.

14 Van Aken, J.R., Novak M., "Curve-Drawing Algorithms for Raster Displays, "ACM TOG, 4(2), April 1985, 147-169.

15 Petráček, P., Vlk, B. & Švéda, J. Linear programming feedrate optimization. *Int J Adv Manuf Technol* **120,** 3625–3646 (2022). https://doi.org/10.1007/s00170-022-08708-0

16 Farouki RT (2008) Pythagorean-hodograph Curves. Springer-Verlag, Berlin, Heidelberg, https://doi.org/10.1007/978-3-540-73398-0_17.

17 Rida T. Farouki, Takis Sakkalis, Real rational curves are not 'unit speed', Computer Aided Geometric Design, Volume 8, Issue 2, 1991, Pages 151-157, ISSN 0167-8396, https://doi.org/10.1016/0167-8396(91)90040-I, (https://www.sciencedirect.com/science/article/pii/016783969190040I)
Abstract: We prove that it is impossible to parameterize any real plane curve, other than a straight line, by rational functions of its arc length. The proof extends in a straightforward manner to real space curves of any dimension d ≥ 3.

18 K. V. Zimenko, et al., (2019),"Interpolation Algorithm for High-Speed Processing of complex curvilinear Trajectories.", November 2019, Conference of open Innovations Association (FRUCT), DOI: 10.23919/FRUCT48121.2019.8981534.

19 Suk-Hwan Suh, S. K. Kang, D.H. Chung, I. Stroud, "Theory and Design of CNC systems", Springer Series in Advanced Manufacturing, 2008, Springer-Verlag, Book DOI 10.1007/978-1/84800-336-1.

20 Caravantes Jorge, Diaz-Toca Gema M. , Fioravanti Mario, Gonzalez-Vega Laureano, "On the Implicit Equation of Conics and Quadrics Offsets," Mathematics (IF2.258), Pub Date : 2021-07-28, DOI: 10.3390/math9151784

21 Huypens V., "Flowchart Berserkless Midpoint Algorithm version 4", https://www.researchgate.net/profile/Valere-Huypens/publications

22 Huypens, Valere (2022), "Mordell the algebraic cubic with offset and constant feedrate", Mendeley Data, V1, doi: 10.17632/5y2d83yygb.1