Perfect 3D-curve RMDPL-IPOs & Bresenham's 3D-Curve Algorithm (Part 2 & 3

Valere Huypens

Perfect 3D-curve RMDPL-IPOs & Bresenham's 3D-Curve Algorithm (Part 2 & 3)

Abstract

The paper presents three new 3D constant feedrate reference pulse interpolations algorithms (RpIPOs or IPOs) that can be used in practical situations in CNC machining tools. The accuracy of the imperfect IPOs is much less (e.g. 37% worse) than the accuracy of the perfect IPOs. The simplified state diagram computes one perfect major axis points and possibly a perfect non-major axis point. The curve algorithms can have singular points, but that problem is simple solved. Each curve is a sub-segment of a monotonic curve from the starting extreme point to the ending extreme point. All the extreme points and the singular points are offline precomputed as the intersection points of three quadrics.
The real time calculation of the arc length of 3D-curve RpIPOs was until now an open problem. As for 2D- and 3D-lines, cascading an ultra-fast 3-lines algorithm "PRM-cs" to the current 3D-curve RpIPO quickly converts this IPO to a constant feedrate RpIPO. The PRM-cs can even be used in integer form and is indeed a real time length algorithm.
The constant feedrate of sampled-data curves is clear when the arc length is known but else impossible. The feedrate, accuracy, flexibility, and execution speed of the new RpIPOs is much better and smoother than those of the sampled-data algorithms. The implementation of the PRM-cs to a 26-connected curve with high accuracy turns out to be a piece of cake in contrast to the sampled-data curves. Furthermore, all IPOs can be converted to real time constant feedrate listSIM-IPOs which can be integrated into rigid simplified CNC machine tools.
The paper generates discrete QSICs in a unique way comparable with 3D-lines thanks to the Relative Minimal Distance of two candidate points to the Polar Line (RMDPL) of the QSIC with respect to the midpoint of the candidate points. The consequences of this theorem are crucial, it means that one must not calculate the time-consuming distance to the 3D curve, but it suffices to calculate the RMDPL. Theoretical, the RMDPL is fundamental, it is the core of all the successful 2D Bresenham algorithms and the paper proves that it is the core of the 3D RpIPOs. The paper gives the codes in CDF-format and explains the algorithms of the next three RpIPOs:
1. The 100% incremental perfect 3D line IPO {49, 74, 82}.
2. The perfect 3D curve IPO of sphere and a cylinder with one singular point.
3. Bresenham's major axis imperfect 3D curve IPO uses the projections of the RMDPL on the non-major axis planes, therefore it is less accurate but super-fast and it can be used in many practical situations as the maximum error (MaxErr) is bounded to 0.707.

Keywords

3D-Bresenham's algorithms · 3D-interpolation · 3D-reference Pulse IPOs · Best 3D-curve IPOs · Best 3D-line IPO · Constant feedrate CNC-interpolation

Abbreviations

Alaska	programming language Alaska Xbase++ version 1.9 from Alaska Software, Germany
BRES-IPO	the collective noun of the MIDP- and two-point method-IPO
26-Connected	each discrete point in a cubic grid has 26 candidate points when the direction is not given and 7 candidate points when the direction is given
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 with tangent parallel to a main axis
RpIPO or IPO	reference pulse or incremental step interpolation algorithm
listSIM	array containing the best points $\mathbf{P}_n$ and the associated length npuls, can be used in real time in rigid simplified CNC machine tools.
LSD	least square distance
LSD-IPO	the collective noun of the VIRT- IPO
Mathematica	the technical computing system of Wolfram Research Illinois, version 10.0.2
MaxErr	the maximum error of the discrete points for a curve
MIDP-IPO	IPO based on the RMDPL method
NURBS	non-uniform rational B-spline
OoC	out-of-control [2, Chapter 6]
OoA	out-of-accuracy [2, Chapter 7]
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
PRMS-cs	converts the LSD-IPOs to constant feedrate IPOs
PRM-IPO	major axis IPO without division or multiplication
QSIC	intersection curve of two quadrics
RMDPL	the Relative Minimal Distance of two candidate points to the Polar Line of the QSIC, quadric or conic with respect to the midpoint of the candidate points
Singular	an extreme point without any direction
VIRT-IPO	IPO based on selecting the candidate point with the LSD with integrated priority

Mathematical Style

The index of new equations is (Eq number) and the equations are referenced as (number). The references are referenced as [number] or [number ; number] or [number–number] and as [number, § paragraph index].
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. Wolfram Language is stored in .nb or .cdf format [35 – 37]. Programs written with Mathematica are in .nb format and need “Mathematica”. They can be converted to .cdf format and you can open them with “Wolfram CDF Player”, but you cannot change them. To change them, you must open them with “Mathematica”.

1. Introduction and overview

1.1. Introduction to the intersection curve of two quadrics (QSIC)

According to [7, § 4.1.2.,Table 5], "*Among all categories, the "Process System" category and "Numerical Control System" category contains the largest number of research concerns, accounting for 34.8% and 19.0% of the total research concerns respectively. According to domain knowledge, these two categories are the focuses of the research in the machine tool domain*". This paper is the sequel of [1;4] and belongs to the "Process System" category with the topics indexes 47 and 85 (interpolation), but with the indexes 10, 16, 22, 63, 83, 121, 125, 128, 136, the paper belongs to the category "Numerical Control system", too. The meaning of the topics and the high-frequent terms of the topics are given in [7, Appendix A, Table A1]. Although in 1981, Koren used the term "reference pulse circle interpolator", the term is not found in [7]. It has always been a problem to find the right term for these interpolators, which are in fact incremental step algorithms (but the term incremental is also not found in [7]). Computer graphics and displays also applied/apply the reference pulse algorithms, but they do not use the name "IPOs", but they call it raster graphics or scan-conversion. The most known are the Bresenham algorithms, the two-point method, the grid-distance method, and the midpoint algorithm [8; 9; 12; 13; 14; 30; 33; 34]. Part 1 of [1; 4] showed that the reference-pulse IPOs have high accuracy, constant feedrate and that high-speed machining can be obtained without using a high-performance CPU. The paper considered 2D-lines, imperfect 3D-lines, conic-curves and 2D-NURBS. The paper also proved that the constant feedrate reference IPOs are much better than the existing sampled-data IPOs: the constant feedrate is better, the accuracy is better, the curves are smoother, the incremental step algorithms calculate the length (npuls) of the curve in real time and the implementation is simpler. This paper combines part 2, the perfect 3D-line IPO and part 3, the perfect 3D-QSIC IPO, because these IPOs are based on the same concept and equations (§ 2). The paper restricts the 3D-curve to QSICs, but that is not a real limitation as quadrics and QSICs are basic objects of CAD/CAM Systems. As the maximum error of all 2D- or 3D-IPOs, with incremental step size Δ, is respectively bounded to 0.5 Δ or 0.707 Δ, the three conditions of [1;4] to apply RpIPOs remain

- the small execution step Δ,
- the small execution time per step,
- the constant feedrate along the curve.

The number of papers about QSICs is large [16 - 29]: Morphology and classification of the QSICs, literature for computing QSICs using the geometric approach for mostly natural quadrics and the algebraic approach producing a (rational) parameterization of the QSIC, tracing QSICs using mostly step marching and some limited cases with interpolation, finding a point of the QSIC, solving the undefined direction in singular points. Determining the nonzero direction vector in a singular point has been solved by [21, pp. 24], [22, pp. 179], and requires the solution of the characteristic equation [22, pp. 195-196, Sec. 8.4], [23, pp. 295 Fig. 3.1], [24, pp. 113, example 6.2]. The singular point is an extreme point, and it is only a tracing problem when the curve starts in the singular point $\mathbf{P}_S$ (no direction); a very practical and simple solution is given (§ 2.6).
The QSIC of the quadrics f1[**P**]=0 and f2[**P**]=0 is identical with the QSIC of one of the quadrics and its pencil, but for machining, one of the quadrics cannot be replaced with a pencil quadric because the geometric surfaces of quadrics are important. The 3D-line is the intersection of two planes or the intersection of two orthogonal planes.
This paper gives not an overview of the existing tracing methods, because the paper represents the QSIC in a unique way as the vector function $\mathbf{f}[\mathbf{P}]=\mathbf{T}[\mathbf{P}]\times\mathbf{P}+\mathbf{W}[\mathbf{P}]=0$ with $\mathbf{T}[\mathbf{P}]$ the tangent vector of the QSIC in point $\mathbf{P}$ and $\mathbf{W}[\mathbf{P}]$ as $-\mathbf{T}[\mathbf{P}]\times\mathbf{P}_S$ and $\mathbf{T}[\mathbf{P}]\bullet\mathbf{P}_S\equiv 0$ such that

$\mathbf{f}[\mathbf{P}]=\mathbf{T}[\mathbf{P}]\times(\mathbf{P}-\mathbf{P}_S)=0$ with $\mathbf{P}_S=\dfrac{\mathbf{T}[\mathbf{P}]\times\mathbf{W}[\mathbf{P}]}{\mathbf{T}[\mathbf{P}]\bullet\mathbf{T}[\mathbf{P}]}$ (27), and for QSICs $\mathbf{W}[\mathbf{P}]$ equals

$S_L*(W_1[\mathbf{P}]*\mathbf{G}_2[\mathbf{P}]-W_2[\mathbf{P}]*\mathbf{G}_1[\mathbf{P}])$ (18) and $\mathbf{T}[\mathbf{P}]$ equals $S_L*(\mathbf{G}_1[\mathbf{P}]\times\mathbf{G}_2[\mathbf{P}])$ (15) with $\mathbf{G}_1[\mathbf{P}]$ and $\mathbf{G}_2[\mathbf{P}]$ the gradients of the quadrics.

Hence, as the equation of the 3D-line is $\mathbf{f}[\mathbf{P}]=\mathbf{L}\times(\mathbf{P}-\mathbf{P}_S)=0$, you would not be surprised that the Bresenham's 3D-line IPO can be extended to Bresenham's 3D-curve IPO (§ 5).
This paper discretizes the QSIC in a constant feedrate 26-connected curve. The geometric tool

DGtal 1.2 can also generate 26-connected curves from 3D parametric curves. DGtal gives no constant feedrate curves, and the tool has connectivity problems when the curvature is high and with large step values. To our knowledge, we found no papers using step interpolation of QSICs or reference pulse interpolation of QSICs with constant feedrate. Reference [11] uses canonical equations, but the MaxErr and the algorithm are unclear to me. The "Space curve interpolation for CNC machines" [38] creates a 26-connected curve (QSIC) with MaxErr bounded to Δ using Danielsson's criteria [38]. It is not a constant feedrate IPO, but the constant feedrate PRM-cs algorithm can easily be integrated. The criterion [38, (9)] in our terminology (§ 2.1) is

- $\mathbf{dP} \triangleq \mathbf{P}_u - \mathbf{P}_A$ (50) (Fig. 1, $\mathbf{P}_u$ is a candidate point, $\mathbf{P}_A$ is the previous best point),
- Maximize $\mathbf{T}[\mathbf{P}_A]\bullet\mathbf{dP}$ corresponds roughly with the priority rules [1, §4.1],
- The selection criterion is $(\mathbf{G}_1[\mathbf{P}_A]\bullet\mathbf{dP})*f_1[\mathbf{P}_A] \le 0$ and $(\mathbf{G}_2[\mathbf{P}_A]\bullet\mathbf{dP})*f_2[\mathbf{P}_A] \le 0$.

  Using (24) with midpoint $\mathbf{P}_A$ this becomes $(f_{L1}[\mathbf{P}_u] - f_{L1}[\mathbf{P}_A])*f_{L1}[\mathbf{P}_A] \le 0$ and $(f_{L2}[\mathbf{P}_u] - f_{L2}[\mathbf{P}_A])*f_{L2}[\mathbf{P}_A] \le 0$ with $f_{L1}[\mathbf{P}_A] \equiv f_1[\mathbf{P}_A]$ and $f_{L2}[\mathbf{P}_A] \equiv f_2[\mathbf{P}_A]$ (24).
  Hence the selected $\mathbf{P}_u$'s are such that $|f_{L1}[\mathbf{P}_u]| \le |f_1[\mathbf{P}_A]|$ and $|f_{L2}[\mathbf{P}_u]| \le |f_2[\mathbf{P}_A]|$, which approximates "each step must drive both f_1 and f_2 towards 0". Then $\mathrm{Max}[\mathbf{T}[\mathbf{P}_A]\bullet\mathbf{dP}]$ selects the best step out of the selected $\mathbf{P}_u$'s. So, the selection is not a distance nor a midpoint criterium, therefore the accuracy is the weak point of this criterium. Bresenham's 3D-curve (§ 5) constant feedrate IPO is faster, its accuracy is better, and the implementation is easier.
  Knuth's 2D curve--algorithm [6] uses the same reasoning "each step must drive inside points to the outside and outside points to the inside". Paper [2, Appendix 3] proves that Knuth's algorithm is identical with the MIDP-algorithm (1; 2; 3), but Danielsson's 2D algorithm is not identical with the MIDP-algorithm, and that is a fortiori valid for his 3D-algorithm.
  The MaxErr of the 3D-line example [39, TABLE IV, pp. 791-792] equals 1.09416 > Δ, when the table is converted to a 26-connected 3D-line. The MaxErr of that line using one of the 3D-line IPOs of [1, § 3.2] or the perfect 3D-line (§ 2) equals 0.543852. As this paper and [1] only considers 26-connected IPOs bounded to 0.707, the IPOs of [39] are not considered.

Recently, [10] adds a fifth imperfect 3D-line IPO to the imperfect-3D-line IPOs of [1; 4, §3.2], it argues to be better than [8], because the "discriminant contains decimals". That is wrong, because the four referenced IPOs [1; 4,§3.2] are basic integer arithmetic algorithms, although most of them can be used with decimal fixed-point arithmetic too. The algorithm [10, Fig. 5] is much slower than the 3D-PRM-IPO of [1;4, § 3.2, Item 3].
Paper [15] references [31], a book of 2008. The statement [15, II Related Work] of the authors of [15] : "*In the first group mentioned* (the Reference-Pulse Interpolators)*, a computer generates reference pulses as an external interrupt signal. The produced pulses are relegated directly to the machine drive. It can achieve high accuracy but as the velocity on each axis depends on external interrupt signal frequency, high speed machining cannot be obtained, and a high-performance Central Processing Unit (CPU) is required*", is false. The 2nd group, the "Reference-Word Interpolators", belongs to the sampled-data IPOs and [1; 4] proved that the reference pulse interpolators are faster, simpler, have much better accuracy and constant feedrate than the sampled-data interpolators. The low cost PIC32 of Microchip is outstanding suitable to realize a real time 3D-curve CNC controller with constant feedrate and using reference-pulse interpolators.

Fig. 1

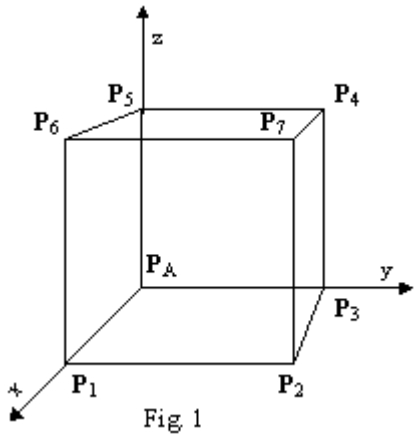


Fig. 1

The 3D-algorithms of this paper, called incremental step IPOs or shortly IPOs are unique (new concept), fast and simple. The curve is always divided into monotonic sub curves with the use of the offline-calculated extreme points (§ 2.6). The paper assumes that [2] and especially the "Relative Curve Measurement Theorem" [2, §5] can be extended to quadrics and QSICs (all experiments show that it can). The consequences of this theorem are not well understood, it means that one must not calculate the time-consuming distance to the 3D-curve for each of the seven candidate points (Fig.1), but it suffices to calculate the Relative Minimal Distance of two candidate points to the Polar Line of the QSIC with respect to the midpoint of the candidate points ("RMDPL") (§2.2; §2.5). The word "polar" induces equations in "polar coordinates", but it has nothing to do with polar coordinates. The polar of the circle $x^2 + y^2 - R^2 = 0$ with respect to the point $\mathbf{P}_m = \{x_m, y_m\}$ is the line $x_m * x + y_m * y - R^2 = 0$. The point $\mathbf{P}_m$ is the pole and by construction is it the midpoint of two candidate points. As the startpoint $\mathbf{P}_A$ of the candidate point is the best point of the previous candidate points, all the midpoints are near to the curve. Hence all the polar lines are near to the curve. The polar line with respect to the startpoint $\mathbf{P}_A$ is not used to select the best point, but the tangent in $\mathbf{P}_A$ is used as the direction vector of the PRM-cs algorithm and the determination of the major axis. So, **the polar lines with respect to the midpoints enclose and inclose the curve**. Experimentation shows that even the absolute distance to the polar lines with respect to the candidate points generates a fairly fitting 26-connected curve, but the value of MaxErr is $> \sqrt{2}/2$, and therefore, by definition, it are bad IPOs, although the mean error is about 0.4 and the standard deviation is smaller than 0.4.
The selection of the best candidate point using the polar line is simple too (44 - 51)
The core of the reference-pulse IPOs is the flexible MIDP-algorithm [2; 1 or 4 §1.1]. Instead of using the virtual Least Squared Distances (LSD) to the **curve**, the MIDP-algorithm uses the **R**elative **M**inimal **D**istances to the **P**olar **Line** (**RMDPL**), as well for the 3D-line as for the 3D-curve. The LSD selects the minimal absolute distance of one of the seven 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), ordered along decreasing priority (44 - 51), or it calculates the minimal absolute distance pairwise.

- The LSD always uses the absolute distance $|\boldsymbol{\rho}_C[\mathbf{P}_u]|$ or the squared distance $\rho_C^2[\mathbf{P}_u]$ $\triangleq \boldsymbol{\rho}_C[\mathbf{P}_u] \bullet \boldsymbol{\rho}_C[\mathbf{P}_u]$ to the curve of one of the candidate points $\mathbf{P}_u$ (30;31, Virtual algorithm).
- The RMDPL (51) selects, pairwise, the minimal squared distance $\text{Min}[\rho_L^2[\mathbf{P}_u], \rho_L^2[\mathbf{P}_v]]$ $\Leftrightarrow$ Select $\mathbf{P}_u$ when $D[\mathbf{P}_u, \mathbf{P}_v] = (\boldsymbol{\rho}_L[\mathbf{P}_u] - \boldsymbol{\rho}_L[\mathbf{P}_v]) \bullet (\boldsymbol{\rho}_L[\mathbf{P}_u] + \boldsymbol{\rho}_L[\mathbf{P}_v]) \le 0$ (41);
  $\rho_L^2[\mathbf{P}_u]$ is the squared distance of the point $\mathbf{P}_u$ to the polar line of the QSIC with respect of the midpoint $\frac{\mathbf{P}_u + \mathbf{P}_v}{2}$ (37). As seen from (40;37), $d[\mathbf{P}_u, \mathbf{P}_v]$ uses the gradient of the midpoint of the two candidate points, therefore it is very flexible, and it allows to calculate the problematic curvatures in real time.

The LSD algorithms are virtual because they consume too much time, the RMDPL algorithms are amazingly fast and for lines the RMDPL's are precomputed constants (§ 3, (76); (77); (78); (79)).

When you analyses the 2D-Bresenham's and its corresponding midpoints algorithms [13;14;30]], they all can be replaced with RMDPL-algorithms (Appendix 7.1).
It seems that the core of the MIDP-algorithm is not well understood, therefore the following simple 2D-clarification of [2].
In 2D, e.g. for the points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ (Fig. 1), the points $\mathbf{P}_M$, $\mathbf{P}_H$, $\mathbf{P}_V$ are respectively the midpoints of the edges $\{\mathbf{P}_u,\mathbf{P}_v\}\in\{\ \{\mathbf{P}_3,\mathbf{P}_1\},\ \{\mathbf{P}_2,\mathbf{P}_1\},\ \{\mathbf{P}_3,\mathbf{P}_2\}\ \}$. The polar line of the conic

$f[\mathbf{P}]=\mathbf{G}\bullet\mathbf{P}+W\equiv\{X,Y\}\bullet\{x,y\}+W$ with respect to the point $\mathbf{P}_m=\dfrac{\mathbf{P}_u+\mathbf{P}_v}{2}$ is

$f_L[\mathbf{P}]=\mathbf{G}_m\bullet\mathbf{P}+W_m\equiv X_m*x+Y_m*y+W_m$ and the distance to the point $\mathbf{P}_u$ equals

$\rho_L[\mathbf{P}_u]=\dfrac{X_m*x_u+Y_m*y_u+W_m}{\sqrt{X_m^2+Y_m^2}}$, $|G_m|=Lm\equiv\sqrt{X_m^2+Y_m^2}$ and the distance to the point $\mathbf{P}_v$ equals

$\rho_L[\mathbf{P}_v]=\dfrac{X_m*x_v+Y_m*y_v+W_m}{Lm}$.

The difference of the squared distances $\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]$ equals

$$(\rho_L[\mathbf{P}_u]-\rho_L[\mathbf{P}_v])*(\rho_L[\mathbf{P}_u]+\rho_L[\mathbf{P}_v])=\frac{\mathbf{G}_m\bullet(\mathbf{P}_u-\mathbf{P}_v)*2*(\mathbf{G}_m\bullet\frac{\mathbf{P}_u+\mathbf{P}_v}{2}+W_m)}{Lm^2}.$$

Hence [2, p. 20, 6 Incremental equation]

$$\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]=2*\frac{\mathbf{G}_m\bullet(\mathbf{P}_u-\mathbf{P}_v)*(\mathbf{G}_m\bullet\mathbf{P}_m+W_m)}{Lm^2}=\frac{1}{Lm^2}*(f_L[P_u]-f_L[P_v])*f_m[P_m].$$

$\mathbf{G}_m=S_{LEFT}(\mathbf{T}_m\times\mathbf{k})=S_{LEFT}(T_{mi}*\mathbf{i}\times\mathbf{k},T_{mj}*\mathbf{j}\times\mathbf{k})=S_{LEFT}\{T_{mj},-T_{mi}\}$ [2, p. 19, eq14] .

The primary OoC conditions [2, § 3, p. 25, (a, b, c)] demand that $\{T_{mj},-T_{mi}\}\equiv\{S_y|T_{mj}|,-S_x|T_{mi}|\}$.

Hence $\mathbf{G}_m=\{X_m,Y_m\}=S_{LEFT}\{S_y|T_{mj}|,-S_x|T_{mi}|\}$.

As $\mathbf{P}_u-\mathbf{P}_v$, equals respectively $\{-S_x,S_y\}\Delta$, $\{0,S_y\}\Delta$ or $\{-S_x,0\}\Delta$

the sign of the difference of the squared distances $\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]$ equals

$Sign[\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]]\equiv Sign[-S_{LEFT}*S_x*S_y*f_m[P_m]]=Sign[-S_{Lxy}*f_m[P_m]]$.

Hence for $\{\mathbf{P}_u,\mathbf{P}_v\}\in\{\ \{\mathbf{P}_3,\mathbf{P}_1\},\ \{\mathbf{P}_2,\mathbf{P}_1\},\ \{\mathbf{P}_3,\mathbf{P}_2\}\ \}$ the relative measurements become

- For $\{\mathbf{P}_u,\mathbf{P}_v\}=\{\mathbf{P}_3,\mathbf{P}_1\}$, $\mathbf{P}_m=\mathbf{P}_M=\dfrac{\mathbf{P}_3+\mathbf{P}_1}{2}$, $Sign[\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]]=Sign[-S_{Lxy}*f_M[P_M]]$, (Eq 1)
- For $\{\mathbf{P}_u,\mathbf{P}_v\}=\{\mathbf{P}_2,\mathbf{P}_1\}$, $\mathbf{P}_m=\mathbf{P}_H=\dfrac{\mathbf{P}_2+\mathbf{P}_1}{2}$, $Sign[\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]]=Sign[-S_{Lxy}*f_H[P_H]]$, (Eq 2)
- For $\{\mathbf{P}_u,\mathbf{P}_v\}=\{\mathbf{P}_3,\mathbf{P}_2\}$, $\mathbf{P}_m=\mathbf{P}_V=\dfrac{\mathbf{P}_3+\mathbf{P}_2}{2}$, $Sign[\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v]]=Sign[-S_{Lxy}*f_V[P_V]]$. (Eq 3)

The crux of [2, § 5] is the relative curve measurement theorem that says that the difference of the squared distances to the conic equals $\overbrace{(\rho_C^2[\mathbf{P}_u]-\rho_C^2[\mathbf{P}_v])}^{\text{Conic}}\equiv(1-\varepsilon_H)*\tau_H*\overbrace{(\rho_L^2[\mathbf{P}_u]-\rho_L^2[\mathbf{P}_v])}^{\text{polar Line}}$.

When the midpoint measurement is not OoC then $1-\varepsilon_H>0$.

When the midpoint measurement is not OoA then $\tau_H>0$.

OoA can occur when $\mathbf{P}_m$ is INSIDE and extremely near to the conic [2, § 7]. **This means that the RMDPL or the difference of the squared distances to a quadratic curve can be measured by replacing the curve by its polar line, and the rarely OoA-error is smaller than { 0.5, $\sqrt{2}$ /2 } for respectively 2D- or 3D-curves**.

The RMDPL will be applied to the perfect 3D-line (§3) and to the perfect 3D-QSIC (the polar lines of the intersection curve of two quadrics) (§ 4). The RMDPL will also be applied to the imperfect 3D-QSIC, and as a tribute to J. Bresenham it is called the "Bresenham's 3D-curve algorithm" (§ 5). You can regard it as an extension of Bresenham's 3D-line algorithm using the intersection of the two polar planes, called the "QSIC polar line". Bresenham's 3D-curve

algorithm is a major axis algorithm, and it measures two projected distances instead of the real 3D-distance, therefore the accuracy (MaxErr, Cost) is not minimal. The perfect RMDPL-curve and imperfect Bresenham's curve algorithms degenerate to perfect 2D midpoint and Bresenham's algorithms, that means that the "polar line" of the RMDPL-algorithm is the fundamental building block of the successful Bresenham or midpoint algorithms.
All the RMDPL algorithms are constant feedrate IPOs, but the parameters {LMcs, LNcs} of the PRM-cs [1, §2] need more explanation (§ 2.8).
Table 4 of [1] indicates that the computation of one best point is not sufficient to generate the perfect 3D-line. The simplified state diagram (§ 2.9, Table 5) of two successive best points is needed to generate the perfect 3D-lines and QSICs.
The specific problems of 3D-curves, the calculation of the extreme points (starting and ending point) and the singular points are treated in paragraph 2.6.
Part 1 showed that the 3D-IPOs were imperfect, and that the accuracy is much less (e.g. 37% worse) than the accuracy of the perfect 3D-line [1]. Part 1 [1;4, § 7]. showed that all IPOs can be converted to constant feedrate listSIM-IPOs which can be used in real time in rigid simplified CNC machine tools.

1.2 Overview

The parameters of the 3D-line from $\mathbf{P}_S = \{x_S, y_S, z_S\}$ to $\mathbf{P}_E = \{x_E, y_E, z_E\}$ are, the direction vector $\mathbf{T}[\mathbf{P}_s] \triangleq \mathbf{L} = \{L_i, L_j, L_k\}$, the length $LS = \text{Norm}[\mathbf{P}_E - \mathbf{P}_S]$ and the absolute components $\{LI, LJ, LK\} = \text{Abs}[\{L_i, L_j, L_k\}]$.

The primary OoC-conditions [2, §3] demand that the monotone direction parameters $S_n \equiv \{S_x, S_y, S_z\} = \text{Sign}[\mathbf{P}_E - \mathbf{P}_S]$ equal the sign of the tangent vector $\mathbf{T}[\mathbf{P}_s]$, hence $\mathbf{L} = \{Sx * LI, Sy * LJ, Sz * LK\}$ (17). For 3D-lines, in contrast with 3D-curves, $\mathbf{L}$ equals $\mathbf{P}_E - \mathbf{P}_S$.

A line has a gradient perpendicular to the line. A 2D-line has one gradient, and a 3D-line has two gradients which define two planes, and the 3D-line is the intersection of the two planes. The line can be defined in two ways.

- Firstly, the distance of a point $\mathbf{P}_n = \{x_n, y_n, z_n\}$ to the 3D-line can be determined from the direction vector $\mathbf{L} = \{L_i, L_j, L_k\} \triangleq \mathbf{P}_E - \mathbf{P}_S$ using the parametric equation $\mathbf{P} = \mathbf{P}_S + \mathbf{L} * t$ equivalent to $\mathbf{f}[\mathbf{P}] = \mathbf{L} \times \mathbf{P} + \mathbf{W} = 0$ and $\mathbf{W} = -\mathbf{L} \times \mathbf{P}_S$ without using the gradients or the residues of the two implicit plane equations. In vector form, $\mathbf{T}[\mathbf{P}_m] \equiv \mathbf{T}[\mathbf{P}_S]$ hence, these equations become $\mathbf{f}[\mathbf{P}] = \mathbf{T}[\mathbf{P}_m] \times \mathbf{P} + \mathbf{W}[\mathbf{P}_m] = 0$ (26), $\mathbf{W}[\mathbf{P}] \triangleq -\mathbf{L} \times \mathbf{P}_S$ (28).
  QSIC polars can also be written in this form with a non-constant $\mathbf{P}_s$ such that $\mathbf{T}[\mathbf{P}_m] \bullet \mathbf{P}_S \equiv 0$ and $\mathbf{P}_S[\mathbf{P}_m] = \dfrac{\mathbf{T}[\mathbf{P}_m] \times \mathbf{W}[\mathbf{P}_m]}{\mathbf{T}[\mathbf{P}_m] \bullet \mathbf{T}[\mathbf{P}_m]}$ (27).
  Hence, the QSIC polar may be written, analogous with the 3D-line, as $\mathbf{P} = \mathbf{P}_S[\mathbf{P}_m] + \mathbf{T}[\mathbf{P}_m] * t$.
- Secondly the 3D-line is the intersection of two 3D-planes $f_1[\mathbf{P}] = \mathbf{G}_1 \bullet \mathbf{P} + W_1[\mathbf{P}] = 0$, $f_2[\mathbf{P}] = \mathbf{G}_2 \bullet \mathbf{P} + W_2[\mathbf{P}] = 0$. Multiplying the former equation with $S_L\mathbf{G}_2$, the latter with $S_L\mathbf{G}_1$ and subtracting gives $\mathbf{f}[\mathbf{P}] = \mathbf{L} \times \mathbf{P} + \mathbf{W}[\mathbf{P}] = 0$ with $\mathbf{L} = S_L * (\mathbf{G}_1 \times \mathbf{G}_2)$, $\mathbf{f}[\mathbf{P}] = S_L * (\mathbf{G}_2 * f_1[\mathbf{P}] - \mathbf{G}_1 * f_2[\mathbf{P}])$, $\mathbf{W}[\mathbf{P}] = S_L * (\mathbf{G}_2 * W_1[\mathbf{P}] - \mathbf{G}_1 * W_2[\mathbf{P}])$ (21;26). The scalar parameter S_L equals ±1 and the sign is such that $S_L * (\mathbf{G}_1 \times \mathbf{G}_2)$ corresponds with the $\mathbf{L}$ or $\mathbf{T}[\mathbf{P}]$ (15;14). So, the second form is as the first form, but $\mathbf{T}[\mathbf{P}]$ and $\mathbf{W}[\mathbf{P}]$, are not directly given, but must be calculated giving the gradients and the two (degenerated) quadrics.

The perfect 3D-line algorithm or the best 3D-line IPO (§ 3) will use the first form, because $\mathbf{f}[\mathbf{P}] = \mathbf{L} \times \mathbf{P} + \mathbf{W}[\mathbf{P}]$ (21) are linear equations of $\mathbf{P}$ and the precalculated coefficients are constants. The best RMDPL point of the seven candidate points (Fig. 1) is found by pairwise comparing two candidate points and the perfect 3D-line IPO needs 21 decision variables (53). Each decision variable is a linear equation with constant coefficients, hence that IPO needs no multiplications, but only additions / subtractions and comparisons with zero. Paragraphs §2.6 and §2.8 show that one can only apply the constant feedrate algorithm, when a major axis move

occurs.
Selecting the best point of the seven candidate points is not sufficient to find the discrete curve with the minimum MaxErr and the minimum Cost, but the state diagram (§2.8) gives the solution.
The IPO of the QSIC, e.g. the intersection of a sphere and a cylinder will use the second form.
This form can be used for the IPO of the perfect 3D-curve or the perfect 26-connected QSIC. The VIRT-IPO (34-35) uses the LSD method [1, §4.4] to compute the parameters MaxErr and Cost.
The $\mathbf{P}_S$ of the equivalent first form always changes; but (47) is the reason that the algorithm of a QSIC cannot be incremental (only additions or subtractions) for 100%.
In addition to the perfect 3D-curve IPO, this paper gives the fast imperfect Bresenham's 3D-curve IPO (§ 5).

2. The QSIC and its polar line equation and the relative decision function

2.1 The QSIC equation in vector form

For each index r = {1,2}, the quadric $f_r[\mathbf{P}]$ is a quadric of the 2nd degree with rational coefficients $\{A_r, B_r, C_r D_r, E_r, F_r, I_r, J_r, M_r\}$, that. can be converted to integers.
The QSIC is the intersection of two quadrics $f_1[\mathbf{P}] \equiv 0$ and $f_2[\mathbf{P}] \equiv 0$ and the IPO must generate the 26-connected curve from $\mathbf{P}_S$ = Round[Startpoint] to $\mathbf{P}_E$ =Round[Endpoint].
The permitted moves are generally Sδ = { δx, δy, δz } = {1, 1, 1 } (Eq 4)
The monotonic direction vector (57) equals

- $S_n = \{S_x, S_y, S_z\} = \text{Sign}[\mathbf{P}_E - \mathbf{P}_S]$. (Eq 5)

The equations of the intersections of the two quadrics become,

- $\mathbf{P} = \{x, y, z\}$ (Eq 6) belongs to the both quadrics f_r,
- $$f_r[\mathbf{P}] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} A_r & D_r & F_r & I_r \\ D_r & B_r & E_r & J_r \\ F_r & E_r & C_r & K_r \\ I_r & J_r & K_r & M_r \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0, \text{ (Eq 7)}$$
- $f_r[\mathbf{P}] = \mathbf{G}_r[\mathbf{P}] \bullet \mathbf{P} + W_r[\mathbf{P}] = 0$, (Eq 8)
- $\mathbf{G}_r[\mathbf{P}] = \{X_r[\mathbf{P}], Y_r[\mathbf{P}], Z_r[\mathbf{P}]\}$, (Eq 9)
- $X_r[\mathbf{P}] = \{A_r, D_r, F_r\} \bullet \mathbf{P} + I_r$ (Eq 10)
- $Y_r[\mathbf{P}] = \{D_r, B_r, E_r\} \bullet \mathbf{P} + J_r$ (Eq 11)
- $Z_r[\mathbf{P}] = \{F_r, E_r, C_r\} \bullet \mathbf{P} + K_r$ (Eq 12)
- $W_r[\mathbf{P}] = \{I_r, J_r, K_r\} \bullet \mathbf{P} + M_r$ (Eq 13)
- $$S_L \triangleq \text{Sign}\left[\frac{(\mathbf{G}_1[\mathbf{P}_s] \times \mathbf{G}_2[\mathbf{P}_s]) \bullet \mathbf{T}[\mathbf{P}_s]}{(\mathbf{G}_1[\mathbf{P}_s] \times \mathbf{G}_2[\mathbf{P}_s])^2}\right] = \text{Sign}[(\mathbf{G}_1[\mathbf{P}_s] \times \mathbf{G}_2[\mathbf{P}_s]) \bullet \{S_x, S_y, S_z\}], \text{ (Eq 14)}$$
- $\mathbf{T}[\mathbf{P}] = \{T_i[\mathbf{P}], T_j[\mathbf{P}], T_k[\mathbf{P}]\} \triangleq S_L(\mathbf{G}_1[\mathbf{P}] \times \mathbf{G}_2[\mathbf{P}])$, (Eq 15)

  $\{TI[\mathbf{P}], TJ[\mathbf{P}], TK[\mathbf{P}]\} \triangleq \text{Abs}[\mathbf{T}[\mathbf{P}]]$, (Eq 16)
- The primary OoC-conditions [2, §3] demand that
  $TI[\mathbf{P}] \equiv S_x * T_i[\mathbf{P}]$, $TJ[\mathbf{P}] \equiv S_y * T_j[\mathbf{P}]$, $TK[\mathbf{P}] \equiv S_n * T_k[\mathbf{P}]$, (Eq 17)
- $\mathbf{W}[\mathbf{P}] = \{W_i[\mathbf{P}], W_j[\mathbf{P}], W_k[\mathbf{P}]\} = S_L * (W_1[\mathbf{P}] * \mathbf{G}_2[\mathbf{P}] - W_2[\mathbf{P}] * \mathbf{G}_1[\mathbf{P}])$. (Eq 18)

The sign parameter S_L is unique defined from the primary OoC condition (17; 5) and the gradients (9) in the startpoint $\mathbf{P}_s$ (14).
Multiplying respectively f_1 with $S_L\mathbf{G}_2$, f_2 with $S_L\mathbf{G}_1$ and subtracting gives, after applying the identity $(\mathbf{A} \bullet \mathbf{C}) * \mathbf{B} - (\mathbf{B} \bullet \mathbf{C}) * \mathbf{A} \equiv (\mathbf{A} \times \mathbf{B}) \times \mathbf{C}$, (Eq 19)

- $\mathbf{f}[\mathbf{P}] \triangleq S_L * (f_1[\mathbf{P}] * \mathbf{G}_2[\mathbf{P}] - f_2[\mathbf{P}] * \mathbf{G}_1[\mathbf{P}]) = S_L * (\mathbf{G}_1[\mathbf{P}] \times \mathbf{G}_2[\mathbf{P}]) \times \mathbf{P} + \mathbf{W}[\mathbf{P}] = 0$, (Eq 20)
- $\mathbf{f}[\mathbf{P}] = \mathbf{T}[\mathbf{P}] \times \mathbf{P} + \mathbf{W}[\mathbf{P}] = 0$. (Eq 21)

The equation of the QSIC is the complex vector function (21). The scalar components are cubics in {x,y,z} and the solution in implicit form is unknown.
The properties of conics [2, p. 20] can be extended to quadrics. The most important is the update equation or the incremental equation for an arbitrary quadric

$$f_{arb}[\mathbf{P}] = \mathbf{G}_{arb}[\mathbf{P}]\bullet\mathbf{P} + W_{arb}[\mathbf{P}] = 0 \text{, with } \frac{\mathbf{G}_{arb}[\mathbf{P}_{new}] + \mathbf{G}_{arb}[\mathbf{P}_n]}{2} \equiv \mathbf{G}_{arb}[\frac{\mathbf{P}_{new} + \mathbf{P}_n}{2}]$$

- $$f_{arb}[\mathbf{P}_{new}] = f_{arb}[\mathbf{P}_n] + 2*(\mathbf{P}_{new} - \mathbf{P}_n)\bullet\mathbf{G}_{arb}[\frac{\mathbf{P}_{new} + \mathbf{P}_n}{2}] \text{. (Eq 22)}$$

Many terms are quadrics, and to update these terms you can use (20) and the 2D-update [5, Item d. Upd] and the extended 3D-update (§ 5.3). In order that the paper will remain clear, the equations will be written in clear form and not in incremental form.
The relative curve measurement theorem extended to QSICs, assumes that

$$\overbrace{(\rho_C^2[\mathbf{P}_2] - \rho_C^2[\mathbf{P}_1])}^{\textbf{curve} \equiv \text{QSIC}} \equiv (1-\varepsilon_H)*\tau_H * \overbrace{(\rho_L^2[\mathbf{P}_2] - \rho_L^2[\mathbf{P}_1])}^{\text{polar } \textbf{line} \text{ of QSIC}} \text{ . (Eq 23)}$$

So, we do not compute the relative squared distances to a QSIC, but we compute the relative squared distances of two candidate points (§ 2.4) of the polar line of the QSIC with respect to the midpoint $\mathbf{P}_m$ of the two candidate points $\{\mathbf{P}_1, \mathbf{P}_2\}$ (§ 2.2).

2.2 The polar line of the QSIC

The polar of a quadric becomes a polar plane. The polar planes of the quadrics (8) with respect to the point $\mathbf{P}_m$ become,

- $$f_{Lr}[\mathbf{P}] = \mathbf{G}_r[\mathbf{P}_m]\bullet\mathbf{P} + W_r[\mathbf{P}_m] = 0 \text{. (Eq 24)}$$

Multiplying respectively $f_{L1}[\mathbf{P}]$ with $S_L\mathbf{G}_2[\mathbf{P}_m]$, $f_{L2}[\mathbf{P}]$ with $S_L\mathbf{G}_1[\mathbf{P}_m]$ and subtracting gives, after applying the identity (19) the intersection of the polar planes,

$$\mathbf{f}_L[\mathbf{P}] \triangleq S_L*(f_{L1}[\mathbf{P}]*\mathbf{G}_2[\mathbf{P}_m] - f_{L2}[\mathbf{P}]*\mathbf{G}_1[\mathbf{P}_m]) = S_L*(\mathbf{G}_1[\mathbf{P}_m]\times\mathbf{G}_2[\mathbf{P}_m])\times\mathbf{P} + \mathbf{W}[\mathbf{P}_m] = 0 \text{, (Eq 25)}$$

- $$\mathbf{f}_L[\mathbf{P}] = \mathbf{T}[\mathbf{P}_m]\times\mathbf{P} + \mathbf{W}[\mathbf{P}_m] = 0. \qquad \text{(Eq 26)}$$

The equation of the polar line of the QSIC is the vector function (26). The scalar components are linear in {x,y,z}. The next section computes the squared distance to the line (26).

2.3 The squared distance of a point $\mathbf{P_n}$ to the polar line of the QSIC with respect to $\mathbf{P_m}$

There always exists a point $\mathbf{P}_S[\mathbf{P}_m]$ such that $\mathbf{f}_L[\mathbf{P}_S[\mathbf{P}_m]] \equiv 0$. This point is the intersection of the polar line with the plane $\mathbf{T}[\mathbf{P}_m]\bullet\mathbf{P}_S[\mathbf{P}_m] = 0$, with $\mathbf{P}_S[\mathbf{P}_m] = \frac{\mathbf{T}[\mathbf{P}_m]\times\mathbf{W}[\mathbf{P}_m]}{\mathbf{T}[\mathbf{P}_m]\bullet\mathbf{T}[\mathbf{P}_m]}$ (Eq 27) and

$$\mathbf{W}[\mathbf{P}_m] = -\mathbf{T}[\mathbf{P}_m]\times\mathbf{P}_S[\mathbf{P}_m] \text{. (Eq 28)}$$

The virtual startpoint $\mathbf{P}$s depends on the midpoint $\mathbf{P}_m$ and the virtual polar line becomes

$$\mathbf{f}_L[\mathbf{P}] = \mathbf{T}[\mathbf{P}_m]\times(\mathbf{P} - \mathbf{P}_S[\mathbf{P}_m]) = 0 \text{. (Eq 29)}$$

To simplify the equations, we put $\mathbf{T}_m \equiv \mathbf{T}[\mathbf{P}_m]$ and $\mathbf{P}_S \equiv \mathbf{P}_S[\mathbf{P}_m]$. (Eq 30)

The distance vector from a point $\mathbf{P}_n$ to the polar line $\mathbf{f}_L[\mathbf{P}]$=0 equals after applying the identity $(\mathbf{C}\times\mathbf{B})\times\mathbf{A} \equiv (\mathbf{C}\bullet\mathbf{A})*\mathbf{B} - (\mathbf{B}\bullet\mathbf{A})*\mathbf{C}$ with $\mathbf{C}\equiv\mathbf{A}\equiv\mathbf{T}_m\equiv\mathbf{T}[\mathbf{P}_m]$ and $\mathbf{B}\equiv\mathbf{P} - \mathbf{P}_S$,

$$\boldsymbol{\rho}[\mathbf{P}_n] = (\mathbf{P}_n - \mathbf{P}_S) - (\frac{\mathbf{T}_m}{|\mathbf{T}_m|}\bullet(\mathbf{P}_n - \mathbf{P}_S))*\frac{\mathbf{T}_m}{|\mathbf{T}_m|} = \frac{(\mathbf{T}_m\bullet\mathbf{T}_m)*(\mathbf{P}_n - \mathbf{P}_S) - (\mathbf{T}_m\bullet(\mathbf{P}_n - \mathbf{P}_S))*\mathbf{T}_m}{(\mathbf{T}_m\bullet\mathbf{T}_m)} \text{ (Eq 31)}$$

$$\boldsymbol{\rho}[\mathbf{P}_n] \equiv \frac{(\mathbf{T}_m\times(\mathbf{P}_n - \mathbf{P}_S))\times\mathbf{T}_m}{(\mathbf{T}_m\bullet\mathbf{T}_m)} \equiv \frac{(\mathbf{T}_m\times\mathbf{P}_n + \mathbf{W}_m)\times\mathbf{T}_m}{(\mathbf{T}_m\bullet\mathbf{T}_m)} \text{. (Eq 32)}$$

Applying the identity $(\mathbf{C}_1\times\mathbf{T}_m)\bullet(\mathbf{C}_2\times\mathbf{T}_m) \equiv (\mathbf{T}_m\bullet\mathbf{T}_m)*(\mathbf{C}_1\bullet\mathbf{C}_2) - (\mathbf{C}_1\bullet\mathbf{T}_m)*(\mathbf{C}_2\bullet\mathbf{T}_m)$ to the squared distance $\boldsymbol{\rho}[\mathbf{P}_n]\bullet\boldsymbol{\rho}[\mathbf{P}_n]$ with $\mathbf{C}_1 \equiv \mathbf{C}_2 \equiv (\mathbf{T}_m\times\mathbf{P}_n + \mathbf{W}_m)$ and $\mathbf{C}_1\bullet\mathbf{T}_m \equiv \mathbf{C}_2\bullet\mathbf{T}_m \equiv 0$ gives,

$$\boldsymbol{\rho}[\mathbf{P}_n]\bullet\boldsymbol{\rho}[\mathbf{P}_n] = \frac{(\mathbf{T}_m\bullet\mathbf{T}_m)*(\mathbf{T}_m\times\mathbf{P}_n + \mathbf{W}_m)\bullet(\mathbf{T}_m\times\mathbf{P}_n + \mathbf{W}_m)}{(\mathbf{T}_m\bullet\mathbf{T}_m)^2} = \frac{(\mathbf{T}_m\times\mathbf{P}_n + \mathbf{W}_m)\bullet(\mathbf{T}_m\times\mathbf{P}_n + \mathbf{W}_m)}{(\mathbf{T}_m\bullet\mathbf{T}_m)} =$$

$$\boldsymbol{\rho}[\mathbf{P}_n]\bullet\boldsymbol{\rho}[\mathbf{P}_n]=\frac{(\mathbf{T}_m\times\mathbf{P}_n+\mathbf{W}_m)\bullet(\mathbf{T}_m\times\mathbf{P}_n+\mathbf{W}_m)}{(\mathbf{T}_m\bullet\mathbf{T}_m)}. \qquad \text{(Eq 33)}$$

This function cannot be used to compute the maximal error MaxErr of the discrete QSIC or to compute the best point of the candidate points, but the time-consuming "RegionDistance" can be used. All the IPOs in the paper compute the discrete points $\mathbf{P}_n$ using the RMDPL criterion (33 – 34; §2.5), and with these points the real distance dErr[Pn] is calculated and finally MaxErr and the Cost.

Measuring the absolute error or distance to a QSIC using the virtual LSD

- Put $q_1=f_1[\mathbf{P}]$ and $q_2=f_2[\mathbf{P}]$ (7; 8),
- Put $\Re$=ImplicitRegion[q1==0 && q2==0, {x, y, z}], (Eq 34)
- $\mathbf{P}_n$ = { x_n, y_n, z_n } is the best selected point,
- dErr[$\mathbf{P}_n$]:=N[Abs[RegionDistance[$\Re$, $\mathbf{P}_n$]]], (Eq 35)
- Store dErr[$\mathbf{P}_n$] (Eq 36) in the array listErr for each selected best point $\mathbf{P}_n$..

Instead of the RMDPL criterion (51) you can use the virtual criterion and in that case you replace d[$\mathbf{P}_u$, $\mathbf{P}_v$] of (44;47;51;50) with Sign[dErr[$\mathbf{P}_u$]-dErr[$\mathbf{P}_v$]] and $\mathbf{P}_v$ equal to $\mathbf{P}_6$, $\mathbf{P}_4$, $\mathbf{P}_2$, $\mathbf{P}_5$, $\mathbf{P}_3$ or $\mathbf{P}_1$. This **virtual algorithm** seeks the LSD of the candidate points to the QSIC.

2.4 The sign of the relative decision function $D[\mathbf{P}_u,\mathbf{P}_v]$ determines the best point

Define $\boldsymbol{\rho}_u=\boldsymbol{\rho}[\mathbf{P}_u]$, $\boldsymbol{\rho}_v=\boldsymbol{\rho}[\mathbf{P}_v]$ and the dot product of $\boldsymbol{\rho}_u\bullet\boldsymbol{\rho}_u$ as $\boldsymbol{\rho}_u^2$ and $\boldsymbol{\rho}_v\bullet\boldsymbol{\rho}_v$ as $\boldsymbol{\rho}_v^2$.

The relative squared distances of two candidate points $\mathbf{P}_u$ and $\mathbf{P}_v$ to the polar line of the QSIC with respect to the midpoint $\mathbf{P}_m\triangleq\frac{\mathbf{P}_u+\mathbf{P}_v}{2}$ (Eq 37) of the candidate points is,

$$D[\mathbf{P}_u,\mathbf{P}_v]\triangleq\boldsymbol{\rho}_u^2-\boldsymbol{\rho}_v^2\triangleq\boldsymbol{\rho}_u\bullet\boldsymbol{\rho}_u-\boldsymbol{\rho}_v\bullet\boldsymbol{\rho}_v\equiv(\boldsymbol{\rho}_u-\boldsymbol{\rho}_v)\bullet(\boldsymbol{\rho}_u+\boldsymbol{\rho}_v). \text{ (Eq 38)}$$

When the priority of point $\mathbf{P}_u$ is higher or equal than the priority of point $\mathbf{P}_v$ the RMDPL criterion selects $\mathbf{P}_u$ when $\text{Sign}\left[D[\mathbf{P}_u,\mathbf{P}_v]\right]\le 0$, else it selects $\mathbf{P}_v$. (Eq 39)

Hence, the best pairwise selection criterion is:

$\mathbf{P}_n=\mathbf{P}_u$ ($\text{Priority}[\mathbf{P}_u]\ge\text{Priority}[\mathbf{P}_v]$); (Eq 40)

$\mathbf{P}_n=\text{If}\left[\text{Sign}\left[D[\mathbf{P}_n,\mathbf{P}_v]\right]\le 0,\ \mathbf{P}_n,\ \mathbf{P}_v\right]$; (Eq 41)

Using (33) and (38) the relative decision function becomes

$$D[\mathbf{P}_u,\mathbf{P}_v]\equiv\frac{2*(\mathbf{T}_m\times(\mathbf{P}_u-\mathbf{P}_v))\bullet(\mathbf{T}_m\times(\frac{\mathbf{P}_u+\mathbf{P}_v}{2})+\mathbf{W}_m)}{\mathbf{T}_m\bullet\mathbf{T}_m} \text{ (Eq 42) or}$$

$$D[\mathbf{P}_u,\mathbf{P}_v]\equiv\frac{2*(\mathbf{T}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]\times(\mathbf{P}_u-\mathbf{P}_v))\bullet(\mathbf{T}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]\times(\frac{\mathbf{P}_u+\mathbf{P}_v}{2})+\mathbf{W}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}])}{\mathbf{T}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]\bullet\mathbf{T}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]}. \text{ (Eq 43)}$$

As $\mathbf{T}_m\bullet\mathbf{T}_m$ has no influence on the Sign of $D[\mathbf{P}_u,\mathbf{P}_v]$, the practical decision function is

$$d[\mathbf{P}_u,\mathbf{P}_v]\equiv(\mathbf{T}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]\times(\mathbf{P}_u-\mathbf{P}_v))\bullet(\mathbf{T}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]\times(\frac{\mathbf{P}_u+\mathbf{P}_v}{2})+\mathbf{W}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]). \text{ (Eq 44)}$$

The form (44) of the decision variable will be used in this paper.

The scalar form of decision function (44)

Using the polar line equation (26) $\mathbf{f}_L[\mathbf{P}]=\mathbf{T}[\mathbf{P}_m]\times\mathbf{P}+\mathbf{W}[\mathbf{P}_m]$, equation (44) converts to

$$d[\mathbf{P}_u,\mathbf{P}_v]=(\mathbf{f}_L[\mathbf{P}_u]-\mathbf{f}_L[\mathbf{P}_v])\bullet\mathbf{f}_L[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]=(\mathbf{f}_L[\mathbf{P}_u]-\mathbf{f}_L[\mathbf{P}_v])\bullet\mathbf{f}[\frac{\mathbf{P}_u+\mathbf{P}_v}{2}]. \text{ (Eq 45)}$$

Using (25) and (26) equation (45) transforms to the scalar form, with $\mathbf{P}_m\triangleq\frac{\mathbf{P}_u+\mathbf{P}_v}{2}$ (37),

$$\begin{aligned}d[\mathbf{P}_u,\mathbf{P}_v]=\ &S_L(f_{L1}[\mathbf{P}_u]*\mathbf{G}_2[\mathbf{P}_m]-f_{L2}[\mathbf{P}_u]*\mathbf{G}_1[\mathbf{P}_m]-f_{L1}[\mathbf{P}_v]*\mathbf{G}_2[\mathbf{P}_m]+f_{L2}[\mathbf{P}_v]*\mathbf{G}_1[\mathbf{P}_m])\\ &\bullet S_L(f_1[\mathbf{P}_m]*\mathbf{G}_2[\mathbf{P}_m]-f_2[\mathbf{P}_m]*\mathbf{G}_1[\mathbf{P}_m])\end{aligned} \qquad \text{(Eq 46)}$$

This can be written in simple scalar form as equation (47) with

$f_{1m} = f_1[\mathbf{P}_m], f_{2m} = f_2[\mathbf{P}_m], f_{1u} = f_{L1}[\mathbf{P}_u], f_{1v} = f_{L1}[\mathbf{P}_v], f_{2u} = f_{L2}[\mathbf{P}_u], f_{2v} = f_{L2}[\mathbf{P}_v],$

$\mathbf{G}^2_{1m} = \mathbf{G}_1[\mathbf{P}_m] \bullet \mathbf{G}_1[\mathbf{P}_m], \mathbf{G}^2_{2m} = \mathbf{G}_2[\mathbf{P}_m] \bullet \mathbf{G}_2[\mathbf{P}_m], \mathbf{G}_{1m} \bullet \mathbf{G}_{2m} = \mathbf{G}_1[\mathbf{P}_m] \bullet \mathbf{G}_2[\mathbf{P}_m].$

$d[\mathbf{P}_u, \mathbf{P}_v] \equiv f_{1m}(f_{1u} - f_{1v})\mathbf{G}^2_{2m} + f_{2m}(f_{2u} - f_{2v})\mathbf{G}^2_{1m} - (f_{1m}(f_{2u} - f_{2v}) + f_{2m}(f_{1u} - f_{1v}))\mathbf{G}_{1m} \bullet \mathbf{G}_{2m}$. (Eq 47)

In 2D $\mathbf{G}_{2m} \equiv \mathbf{k}$, $\mathbf{G}_{1m} \bullet \mathbf{G}_{2m} \equiv 0$ and f_2 is zero or a constant, hence the 2D decision function (1), (2), (3), becomes the well-known midpoint decision function $df_m[P_u, P_v] \equiv (f_{1u} - f_{1v}) * f_{1m}$.

2.5 The concept of the RMDPL algorithm

The imperfect Bresenham's 3D-curve algorithm (§ 5) is a major axis algorithm, and it does not use the RMDPL criterion as it measures two projected distances instead of the real 3D-distance. The perfect 3D-line and the perfect 3D-curve IPOs use the same concept.

The candidate points (Fig. 1), with starting point $\mathbf{P}_A$, are ordered along decreasing priority

listCan ≡ { $\mathbf{P}_7$, $\mathbf{P}_6$, $\mathbf{P}_4$, $\mathbf{P}_2$, $\mathbf{P}_5$, $\mathbf{P}_3$, $\mathbf{P}_1$ }. (Eq 48)

The pairwise algorithm RMDPL considers two candidate points $\{\mathbf{P}_u, \mathbf{P}_v\} \in$ listCan (Eq 49)

and their midpoint (37) $\mathbf{P}_m \triangleq \frac{\mathbf{P}_u + \mathbf{P}_v}{2}$.

The candidate points are obtained from $\mathbf{P}_A$ and the monotonic direction Sn≡{Sx, Sy, Sz},

$\mathbf{P}_1 = \mathbf{P}_A + \{S_x, 0, 0\}$, $\mathbf{P}_2 = \mathbf{P}_A + \{S_x, S_y, 0\}$, $\mathbf{P}_3 = \mathbf{P}_A + \{0, S_y, 0\}$, $\mathbf{P}_4 = \mathbf{P}_A + \{0, S_y, S_z\}$, (Eq 50)

$\mathbf{P}_5 = \mathbf{P}_A + \{0, 0, S_z\}$, $\mathbf{P}_6 = \mathbf{P}_A + \{S_x, 0, S_z\}$, $\mathbf{P}_7 = \mathbf{P}_A + \{S_x, S_y, S_z\}$.

The RMDPL criterion can use (44) or (47) and both use the next selection algorithm,

Selection algorithm using the RMDPL criterion (Eq 51)

- If [δx==1 && δy==1 && δz==1,

  Use listCan (48) and put $\mathbf{P}_u = \mathbf{P}_7$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_6]] \le 0, \mathbf{P}_u, \mathbf{P}_6]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_4]] \le 0, \mathbf{P}_u, \mathbf{P}_4]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_2]] \le 0, \mathbf{P}_u, \mathbf{P}_2]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_5]] \le 0, \mathbf{P}_u, \mathbf{P}_5]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_3]] \le 0, \mathbf{P}_u, \mathbf{P}_3]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_1]] \le 0, \mathbf{P}_u, \mathbf{P}_1]$;

  ];

- If [δx==1 && δy==1 && δz==0,

  Use listCan={ $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_1$ } and put $\mathbf{P}_u = \mathbf{P}_2$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_3]] \le 0, \mathbf{P}_u, \mathbf{P}_3]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_1]] \le 0, \mathbf{P}_u, \mathbf{P}_1]$;

  ];

- If [δx==1 && δy==0 && δz==1,

  Use listCan={ $\mathbf{P}_6$, $\mathbf{P}_5$, $\mathbf{P}_1$ } and put $\mathbf{P}_u = \mathbf{P}_6$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_5]] \le 0, \mathbf{P}_u, \mathbf{P}_5]$;

  $\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_1]] \le 0, \mathbf{P}_u, \mathbf{P}_1]$;

  ];

- If [δx==0 && δy==1 && δz==1,

  Use listCan={ $\mathbf{P}_4$, $\mathbf{P}_5$, $\mathbf{P}_3$ } and put $\mathbf{P}_u = \mathbf{P}_4$;

$\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_5]] \le 0, \mathbf{P}_u, \mathbf{P}_5\ ];$

$\mathbf{P}_u = \text{If}[\ \text{Sign}[d[\mathbf{P}_u, \mathbf{P}_3]] \le 0, \mathbf{P}_u, \mathbf{P}_3\ ];$

];

- If [δx==1 && δy==0 && δz==0,

  Use listCan={ $\mathbf{P}_1$ } and put $\mathbf{P}_u = \mathbf{P}_1$;

  ];
- If [δx==0 && δy==1 && δz==0,

  Use listCan={ $\mathbf{P}_3$ } and put $\mathbf{P}_u = \mathbf{P}_3$;

  ];
- If [δx==0 && δy==0 && δz==1,

  Use listCan={ $\mathbf{P}_5$ } and put $\mathbf{P}_u = \mathbf{P}_5$;

  ];
- $\mathbf{P}_{min} = \mathbf{P}_u$ or in integer format nMin = nu ((§ 3; (80); (81));

So, the algorithm has six comparisons and applies (44) six-times, but one must update (44) twenty-one-times, because the possible pairwise points enumerate as
listofPairs = { { $\mathbf{P}_7$,$\mathbf{P}_6$}, { $\mathbf{P}_7$,$\mathbf{P}_4$}, { $\mathbf{P}_7$,$\mathbf{P}_2$}, { $\mathbf{P}_7$,$\mathbf{P}_5$}, { $\mathbf{P}_7$,$\mathbf{P}_3$}, {$\mathbf{P}_7$,$\mathbf{P}_1$}, (Eq 52)
{$\mathbf{P}_6$,$\mathbf{P}_4$}, {$\mathbf{P}_6$,$\mathbf{P}_2$}, {$\mathbf{P}_6$,$\mathbf{P}_5$}, {$\mathbf{P}_6$,$\mathbf{P}_3$}, {$\mathbf{P}_6$,$\mathbf{P}_1$}, {$\mathbf{P}_4$,$\mathbf{P}_2$}, {$\mathbf{P}_4$,$\mathbf{P}_5$}, {$\mathbf{P}_4$,$\mathbf{P}_3$}, {$\mathbf{P}_4$,$\mathbf{P}_1$},
{$\mathbf{P}_2$,$\mathbf{P}_5$}, {$\mathbf{P}_2$,$\mathbf{P}_3$}, {$\mathbf{P}_2$,$\mathbf{P}_1$}, {$\mathbf{P}_5$,$\mathbf{P}_3$}, {$\mathbf{P}_5$,$\mathbf{P}_1$}, {$\mathbf{P}_3$,$\mathbf{P}_1$} }
The number of possible pairwise points is Length[listofPairs] = $\binom{7}{2} \equiv 21$. (Eq 53)

2.6 The offline computation of the extreme and singular points

The QSIC with tangent **T**[{x,y,z}] = 0 (15) is the intersection of the two quadrics f1[x,y,z]=0, f2[x,y,z]=0 (7; 8). The extreme points are the tangent points of the QSIC parallel with respectively Ti[x,y,z]=0, Tj[x,y,z]=0 and Tk[x,y,z]=0. Each component of **T**[{x,y,z}] = is a quadric, hence, the extreme points are the intersection of three quadrics f1[x,y,z]=0, f2[x,y,z]=0, (7; 8) and respectively Ti[x,y,z]=0, Tj[x,y,z]=0 and Tk[x,y,z]=0. That intersection problem is much simpler than the general three quadrics intersection problem.
This paper uses NSolve of Mathematica:

- Chop[NSolve[{ f1[x,y,z]==0, f2[x,y,z]==0, **T**[{x,y,z}] [[1]]==0 },{x,y,z},Reals]]; (Eq 54)
- Chop[NSolve[{ f1[x,y,z]==0, f2[x,y,z]==0, **T**[{x,y,z}] [[2]]==0 },{x,y,z},Reals]]; (Eq 55)
- Chop[NSolve[{ f1[x,y,z]==0, f2[x,y,z]==0, **T**[{x,y,z}] [[3]]==0 },{x,y,z},Reals]]; (Eq 56)

The common solutions of (54) –(56) are singular points (rendered as “White bullets”).
The residual extreme points, (“//” means parallel to)

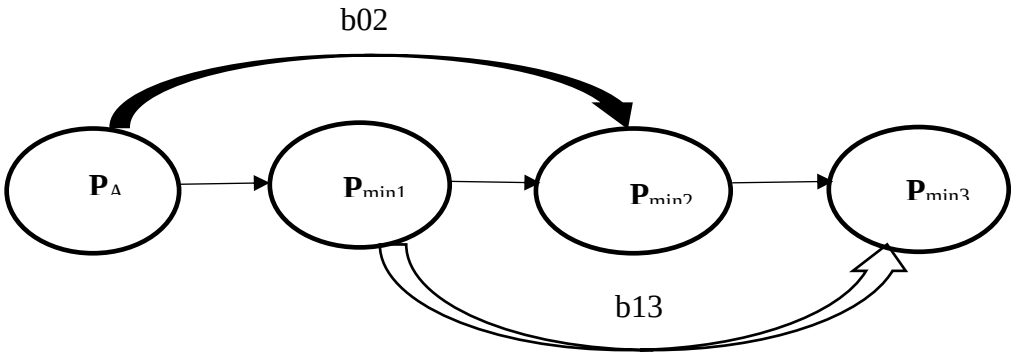
- with {Ti==0, Tj≠0; Tk≠0} are the extremes // Ti=0 (filtered & rendered as “Blue bullets”);
- with {Tj==0, Tk≠0; Ti≠0} are the extremes // Tj=0 (filtered & rendered as “Red bullets”);
- with {Tk==0, Ti≠0; Tj≠0} are the extremes // Tk=0 (filtered & rendered as “Green bullets”);

The 26-connected QSICs are always segments {$\mathbf{P}_S$, $\mathbf{P}_E$} of the monotonic extreme segments {$\mathbf{P}_{ext1}$, $\mathbf{P}_{ext2}$}, therefore the monotonic direction vector always equals
$S_n = \{S_x, S_y, S_z\} = \text{Sign}[\mathbf{P}_E - \mathbf{P}_S] = \text{Sign}[\mathbf{P}_{ext2} - \mathbf{P}_{ext1}]$ (5) . (Eq 57)
Point $\mathbf{P}_E$ can be a singular point, and point $\mathbf{P}_S$ can be an extreme point but not a singular point. Hence, we have a big tracing problem when the curve starts in the singular point $\mathbf{P}_S$ (no direction), but the monotonic direction vector S_n in the singular point is clearly defined. Therefore using the Virtual LSD (30;31, Virtual algorithm), the singular point $\mathbf{P}_S$ and the monotonic direction vector $S_n = \{S_x, S_y, S_z\}$, the possible next best point is the point with the minimal distance of the candidate points listCan (47-52) with $\mathbf{P}_A \equiv \mathbf{P}_S$. Instead of starting in the singular point $\mathbf{P}_S$ the

algorithm starts in the next best point.

2.7 The Simplified State Diagram

Fig. 2: State Diagram of three paths path1, path2, path3



To clearly define the wrong 26-connected paths we take segments of the imperfect 3D-line of [1;4,§ 3.2.2.1 & Table 4].
Table 1, 2 and 3 represent three possible paths for the 3D-line L={49,74,82} and {Sx, Sy, Sz}={δx, δy, δz}={1, 1, 1}.
The used parameters for this line are with $\mathbf{P}_{min1} \equiv \mathbf{P}_{A1}$, $\mathbf{P}_{min2} \equiv \mathbf{P}_{A2}$ and $\mathbf{P}_{min3} \equiv \mathbf{P}_{A3}$,

- the main axis index majN ≡ 3 corresponding with the major z-axis direction in the startpoint $\mathbf{P}_A$,
- the three successive LSD points $\mathbf{P}_{A1}$, $\mathbf{P}_{A2}$ and $\mathbf{P}_{A3}$,
- the Boolean b02 is true when $\mathbf{P}_{A2}$ is 26-connected with $\mathbf{P}_A$,
- the Boolean b13 is true when $\mathbf{P}_{A3}$ is 26-connected with $\mathbf{P}_{A1}$,
- the Boolean bmajAi is true when the move to $\mathbf{P}_{Ai}$ is major axis move ⇔ majN.
- dErr is the absolute error or distance to a QSIC using the virtual LSD (35)

Table 1: of Path1

	Path 1 $\mathbf{P}_A \rightarrow \mathbf{P}_{min1} \rightarrow \mathbf{P}_{min3}$					
Points	Value	dErr	b02	b13	bmajAi	
$\mathbf{P}_A$	{33,50,56}	0.485	False			
$\mathbf{P}_{A1}$	{34,51,57	0.501		True	True	
$\mathbf{P}_{A2}$	{34,52,57	0.466	False		False	
$\mathbf{P}_{A3}$	{35,52,58	0.507		True	True	

Table 2 of Path2

		Path 2 $\mathbf{P}_A \rightarrow \mathbf{P}_{min2} \rightarrow \mathbf{P}_{min3}$				
Points		Value	dErr	b02	b13	bmajAi
$\mathbf{P}_A$		{34,51,57}	0.501	True		
$\mathbf{P}_{A1}$		{34,52,57}	0.466		False	False
$\mathbf{P}_{A2}$		{35,52,58}	0.507	True		True
$\mathbf{P}_{A3}$		{35,55,59}	0.712		False	True

Table 3 of Path3

		Path 3 $\mathbf{P}_A \rightarrow \mathbf{P}_{min1} \rightarrow \mathbf{P}_{min2} \rightarrow \mathbf{P}_{min3}$				
Points		Value	dErr	b02	b13	bmajAi
$\mathbf{P}_A$		{03,04,05}	0.408	False		
$\mathbf{P}_{A1}$		{03,05,05}	0.382		False	False
$\mathbf{P}_{A2}$		{04,06,06}	0.487	False		True
$\mathbf{P}_{A3}$		04,06,07}	0.249		False	True

When b02 is True then points $\mathbf{P}_{A1}$ and $\mathbf{P}_{A2}$ are LSD candidate points of $\mathbf{P}_A$, hence $dErr[\mathbf{P}_{A1}] \le dErr[\mathbf{P}_{A2}]$.
When b13 is True then points $\mathbf{P}_{A2}$ and $\mathbf{P}_{A3}$ are LSD candidate points of $\mathbf{P}_{A1}$, hence $dErr[\mathbf{P}_{A2}] \le dErr[\mathbf{P}_{A3}]$.
Therefore the selected paths are as in Table 4.

Table 4 {b02,b03} State diagram

b02	b13	Paths
False	False	$\mathbf{P}_A \rightarrow \mathbf{P}_{A1} \rightarrow \mathbf{P}_{A2} \rightarrow \mathbf{P}_{A3}$
True	False	$\mathbf{P}_A \rightarrow \rightarrow\rightarrow \rightarrow \mathbf{P}_{A2} \rightarrow \mathbf{P}_{A3}$
False	True	$\mathbf{P}_A \rightarrow \mathbf{P}_{A1} \rightarrow\rightarrow\rightarrow\rightarrow \mathbf{P}_{A3}$
True	True	$\mathbf{P}_A \rightarrow \mathbf{P}_{A1} \rightarrow\rightarrow\rightarrow\rightarrow \mathbf{P}_{A3}$

The non-major axis rule: "**Two successive non-major axis moves are impossible**".
The proper IPO can apply the general state diagram, but the IPO only applies the PRM-cs (feedrate) for each major axis move (§ 2.8). When $\mathbf{P}_{A1}$ is a major axis move (bmajA1=True) the IPO can be restarted, else when $\mathbf{P}_{A1}$ is a non-major axis move (bmajA1=False), the non-major axis rule says that the next best point is a major axis move (bmajA2≡True), and the IPO can then be restarted.

The simplified state diagram applies the non-major axis rule "**Two successive non-major axis moves are impossible**". LSD point $\mathbf{P}_{A3}$ must not be calculated and b13 is always false. The simplified state diagram becomes (Table 5):

Table 5 Simplified state diagram

bmajA1	bmajA2	b02	b13	Paths
True	DC	False	False	$\mathbf{P}_A \rightarrow \mathbf{P}_{A1}$
False	True	True	False	$\mathbf{P}_A \rightarrow \rightarrow\rightarrow \rightarrow \mathbf{P}_{A2}$
False	True	False	False	$\mathbf{P}_A \rightarrow \mathbf{P}_{A1} \rightarrow \mathbf{P}_{A2}$

All perfect IPOs in this paper will use the simplified state diagram.

2.8 The constant feedrate algorithm PRM-cs is also valid for a 3D-curve

The PRM-cs algorithm was proved for 2D-lines [1, §2.1-2, §5.3].
The 3D-polar line (26) equals $\mathbf{f}_L[\mathbf{P}] = \mathbf{T}_m \times (\mathbf{P} - \mathbf{P}_S) = 0$ (29), with $\mathbf{T}_m = \mathbf{T}[\mathbf{P}_m]$ and $\mathbf{P}_S = \dfrac{\mathbf{T}_m \times \mathbf{W}_m}{\mathbf{T}_m \bullet \mathbf{T}_m}$ (27; 30). $\mathbf{P}_m$ corresponds with one of the 21 midpoints (53) when defining the best candidate point, but the feedrate algorithm PRM-cs takes $\mathbf{T}_m$ in the startpoint $\mathbf{P}_A$ of the cubic hence $\mathbf{T}_m \equiv \mathbf{T}_A$ when using the PRM-cs.
The 3D-polar line of a QSIC $\mathbf{T}_m \times (\mathbf{P} - \mathbf{P}_S) = \mathbf{T}_A \times (\mathbf{P} - \mathbf{P}_S) = 0$ is equivalent to

$$\frac{x - x_S}{|T_{Ai}|} = \frac{y - y_S}{|T_{Aj}|} = \frac{z - z_S}{|T_{Ak}|} = \frac{nkey}{MATA} = \frac{npuls}{NATA} = t \ , \ 0 \le t \le 1 \ (npuls \equiv arclength) \text{ with}$$

ATA = Abs[$\mathbf{T}_A$] = { $|T_{Ai}|$, $|T_{Aj}|$, $|T_{Ak}|$ }, (Eq 58)
MATA = Max[ATA], (Eq 59)
NATA = N[Norm[ATA]], (Eq 60)

LN = $2*$Ceiling[Norm[$\mathbf{P}_{monoE} - \mathbf{P}_{monoS}$] + .05] (Eq 61) is the normalized length,

or LN = = Ceiling[$2*$Norm[$\mathbf{P}_{monoE} - \mathbf{P}_{monoS}$] + .05], LN = If[OddQ[LN], LN, LN+1],
kcs = 1 when Floor is not used, (Eq 62)

LM = $\frac{MATA}{NATA} * LN$ (Eq 63) when kcs ≡ 1, else

LM = Floor[$\frac{1}{kcs}\frac{MATA}{NATA} * LN + 0.5$], (Eq 64)

ACS = $\frac{LN}{2}$, (Eq 65)

majN = Switch[MATA, $|T_{Ai}|$, 1, $|T_{Aj}|$, 2, $|T_{Ak}|$, 3]. (Eq 66)

For 3D-curves, the vector $\mathbf{T}_A$ must be normalized to obtain the correct feedrate.
The monotonic startpoint is $\mathbf{P}_{monoS}$ and the monotonic endpoint is $\mathbf{P}_{monoE}$ and its length is rounded to the even normalized length LN (61). LN can be used as the normalized length on the condition that LN is greater than NATA (else double LN).

The PRM-cs is a pulse rate algorithm [1;4 § 5) that calculates nkey = Round[$\frac{MATA}{NATA} * nputs$] or nputs = Round[$\frac{NATA}{MATA} * nkey$]. MATA corresponds with the maximal absolute component of the polar line or with the major axis direction majN of the polar line in $\mathbf{P}_A$.
So, we can only update nputs when the 3D-curve makes a major axis move.
In that case the 3D-PRM-cs does not change and the conclusion of [1, §5.3] is valid, and the 3D-curve PRM-cs becomes .

```
If[major axis move,
  Label[StartCS];
  nputs = nputs +1;
  ACS = ACS – LM;
  If[ ACS > 0, Goto[StartCS]];
  ACS = ACS + LN;
];
```

The PRM-cs is not an integer algorithm when LM uses (63).
The PRM-cs is an integer algorithm when LM uses (64).
The determination of kcs when LM uses (64):

1) Calculate nputs or determine nputs with kcs equal to 1 and without Floor equation (63), and put the value in the numerator of kcs;
2) Calculate nputs with kcs equal to 1 and use the Floor equation (64), and put the value in the denominator of kcs;
3) Calculate nputs with kcs and the floor equation (64).
   Eventually change the denominator with ±0.5.
   e.g. For the 3D-line {49,74,82} the kcs with (64) equals $\frac{120.835}{121}$, but the correct value using (64) equals $\frac{120.835}{120.5}$

Conclusions of § 2.8:

- You can only run the PRM-cs algorithm when the 3D-curve makes a major axis move,
- The length of the vector $\mathbf{T}_A$ must be normalized to obtain the correct feedrate,
- The PRM-cs algorithm can use integer arithmetic.

2.9 Introduction to the examples

The computation of the maximal error MaxErr uses (34; 35) and (36) stores dErr[$\mathbf{P}_n$] in listErr as listErr = Join[listErr, { dErr[Pn] }] and MaxErr = Max[listErr].
The computation of dErr[$\mathbf{P}_n$] is very time consuming (35), and the IPO does not need MaxErr,

therefore the examples will not show that computation, but the calculated MaxErr will be given. The 3D-line will give the real-time values dErr[$\mathbf{P}_n$] because it uses

dErr[P_]:=N[Abs[$\frac{\text{Norm}[L \times P]}{\text{Norm}[L]}$]] (Eq 67) which is less time-consuming than Mathematica's more general RegionDistance function (34; 35).

Mathematica's execution time of reference pulse algorithms is always very bad in comparison with Alaska Xbase++ or C++, e.g. [1; 4; §8.1 Table 8], $\frac{\text{Mathematica's listSIM}}{\text{Alaska's listSIM}} = \frac{132}{0.07} = 1885.71$ or 188471 % slower. Therefore you can only compare Mathematica's reference pulse algorithms among themselves.

Instead of the RMDPL criterion (51) you can use the virtual criterion and in that case you replace d[$\mathbf{P}_u$, $\mathbf{P}_v$] of (51) with Sign[dErr[$\mathbf{P}_u$]-dErr[$\mathbf{P}_v$] and $\mathbf{P}_v$ equal to $\mathbf{P}_6$, $\mathbf{P}_4$, $\mathbf{P}_2$, $\mathbf{P}_5$, $\mathbf{P}_3$ or $\mathbf{P}_1$. This **virtual algorithm** seeks the LSD of the candidate points to the QSIC.

This paragraph gives the results of five examples, with R = 500, r = 400 and z0 = 300,

a. 3D-Line L={49, 74, 82},
b. 3D-QSIC ≡ Intersection of the quadrics $h1 = x^2 + y^2 + z^2 - R^2$ & $g1 = x^2 - R*x + z^2$,
c. 3D-QSIC ≡ Intersection of $h2 = x^2 + y^2 + z^2 - R^2$ and the plane $g2 = 4*x - 3*y$,
d. 3D-QSIC ≡ Intersection of the quadrics $h3 = x^2 + y^2 + z^2 - R^2$ & $g3 = 16*x^2 - 9*y^2$,
e. 3D-QSIC ≡ Intersection of the sphere $h4 = x^2 + y^2 + z^2 - R^2$ & $g4 = z^2 - z0^2$,
   $z0 = \sqrt{R^2 - r^2}$.

The extreme points (§ 2.6) are computed with (54; 55; 56) and the singular points are stored in PtsWHITE, and the other extreme points of Ti=0, Tj=0, Tk=0 in respectively PtsBLUE, PtsRED and PtsGREEN. Mathematica creates images of the intersections b, c, d, and e, their quadrics or degenerate quadrics and their extremes (dot colored) (Table 8, Fig. b1-b3, b4-b5, d1-d3, e2-e3).

Table 6 Extreme points of examples b, c, d, and e

Extremes	QSIC b $h1 \cap g1 = 0$	QSIC c $h2 \cap g2 = 0$	QSIC d $h3 \cap g3 = 0$	QSIC e $h4 \cap g4 = 0$

PtsWHITE	{{R, 0, 0}}	{
		{0, 0, -R},
		{0, 0, R}
		}
			{
			{0, 0, -R},
			{0, 0, R}
			}
				{}

PtsBLUE	{}	{}	{}	{
				{-r, 0, -z0},
				{ r, 0, -z0},
				{-r, 0, z0},
				{ r, 0, z0}
				}

PtsRED	{
	{0, R, 0},
	{0, -R, 0}
	}
		{}	{
			$\{\frac{R}{5}\{3,4,0\}$,
			$\{\frac{R}{5}\{-3,-4,0\}\}$
				{
				{0, -r, -z0},
				{0, r, -z0},
				{0, -r, z0},
				{0, r, z0}
				}

PtsGREEN	{
	$\{1,-\sqrt{2},1\}*\frac{R}{2}$,
	$\{1,\sqrt{2},1\}*\frac{R}{2}$,
		{
		$\{\frac{R}{5}\{-3,4,0\}$,
		$\{\frac{R}{5}\{3,-4,0\}$
		}
			{
			$\{\frac{R}{5}\{-3,4,0\}$,
			$\frac{R}{5}\{3,-4,0\}$
			}
				{}

	$\{1,-\sqrt{2},-1\}*\frac{R}{2}$ $\{1,\sqrt{2},-1\}*\frac{R}{2}\}$ }			

All start and endpoints must be integers, hence $\{1,-\sqrt{2},1\}*\frac{R}{2}$ is rounded to {250, -354, 250} .

Example b creates the QSIC from {250, -354, 250} to the singular point {500, 0,0}. Therefore the paper adds a second example {b4, b5}. The second segment starts in the singular point, therefore the best startpoint calculated with (34; 35) for PA={500,0,0} and the candidate points (50) in listCan (48) with

$\{S_x, S_y, S_z\}$ = Sign[{250, 354, 250}-{500, 0, 0}] = {-1, 1, 1} equals $\mathbf{P}_{S2}$ = {500, 1, 1} (LSD 34;35;36).

The calculated lengths of a, b, c, d, and e are respectively,

a. $\sqrt{\mathbf{P}_E^2-\mathbf{P}_S^2}=120.835\simeq 121,$

b. $\text{ArcLength}\left[R*\left\{\frac{1}{1+t^2},-\frac{\sqrt{2}*t^2}{1+t^2},\frac{t}{1+t^2}\right\},\{t,0,1\},\text{Method}\to\text{"NIntegrate"}\right]=$

$\text{ArcLength}\left[\frac{R}{2}*\{1+\text{Cos}[\theta],-\sqrt{2}*(1-\text{Cos}[\theta]),\text{Sin}[\theta]\},\left\{\theta,0,\frac{\pi}{2}\right\},\text{Method}\to\text{"NIntegrate"}\right]=$

$546.11\simeq 546.$

The parametric representation of the QSIC is

$\{x, y, z\} = \frac{R}{2}*\{1+\text{Cos}[\theta],-\sqrt{2}*(1-\text{Cos}[\theta]),\text{Sin}[\theta]\}.$

c. $\text{ArcLength}\left[R*\left\{\frac{4}{5}\text{Cos}[\theta,\ \frac{3}{5}\text{Cos}[\theta,\ \text{Sin}[\theta]\right\},\left\{\theta,0,\frac{\pi}{2}\right\}\right]=\frac{\pi}{2}R=785.4\simeq 785.$

The parametric representation of the QSIC is

$\{x, y, z\} = R*\left\{\frac{4}{5}\text{Cos}[\theta,\ \frac{3}{5}\text{Cos}[\theta,\ \text{Sin}[\theta]\right\}.$

d. As c.

e. $\frac{\pi}{2}r = 628.319\simeq 628.$

The examples are subdivided into three groups : the Virtual IPO (34-36), the RMDPL-IPO (§ 3; § 4) and the Bresenham's 3D-curve-IPO (§ 5).

The maximal error MaxErr is only once computed and copied to MaxErr when dErr is not computed.

Table 7 Results of the Virtual-, RMDPL- and Bresenham's-IPO for examples a, b, c, d, and e

Properties of Bresenham's 3D-curve algorithm versus the best point algorithm	
IPOs with R = 500, r = 400 a: h1 = $x^2 + y^2 + z^2 - R^2 = 0$, b: h1 & g1 = x^2-R x+z^2 = 0, c: h1 & g2 = 4 x – 3 y = 0, d: h1 & g3 = 16 x^2 – 9 y^2,	All PRM-cs algorithms use the normalized length. The execution time nTime is valid for IPOs written with Mathematica and when MaxErr is not measured (dErr ≡ 0). All IPOs are perfect except Bresenham's -3D.

e: h1 & g4 = $z^2 - z_0^2 = 0$, $z_0 = \sqrt{R^2 - r^2} = 300$.					The value (nTime) is the time when MaxErr is calculated.		
	3D-curve	npuls	MaxErr	nTime	$\mathbf{P}_S$	$\mathbf{P}_E$	Info
a1	3D-Line Virt-IPO	121	0.497124	0.25	{0, 0, 0}	{49, 74, 82}	
a2	3D-Line Bres-3D	121	0.682438	0.04	{0, 0, 0}	{49, 74, 82}	
a3	3D-Line RMDPL	121	0.497124	0.08	{0, 0, 0}	{49, 74, 82}	
b1	h1 & g1 Virt-IPO	546	0.558677	128.4	{250, -354, 250}	{500, 0, 0}	$\mathbf{P}_E$ is singular
b2	h1 & g1 RMDPL	546	0.558677	1.2 (11)	{250, -354, 250}	{500, 0, 0}	$\mathbf{P}_E$ is singular
b3	h1 & g1 Bres-3D	546	0.64686	0.305 (9.7)	{250, -354, 250}	{500, 0, 0}	$\mathbf{P}_E$ is singular
b4	h1 & g1 RMDPL	1092	0.558677	2.4 (21.4)	{250, -354, 250} {500, 1, 1}	{500, 0, 0} {250, 354, 250}	$\mathbf{P}_{E1}$ is singular
b5	h1 & g1 Bres-3D	1092	0.64686	0.61 (19.6)	{250, -354, 250} {500, 1, 1}	{500, 0, 0} {250, 354, 250}	$\mathbf{P}_{E1}$ is singular
d1 c1	h1 & g3 Virt-IPO	785	0.634549	106.6	{300, -400, 0}	{0, 0, 500}	$\mathbf{P}_E$ is singular
d2 c2	h1 & g3 RMDPL	785	0.634549	1.95 (11)	{300, -400, 0}	{0, 0, 500}	$\mathbf{P}_E$ is singular
d3 c3	h1 & g3 Bres-3D	785	0.697586	0.45 (13.7)	{300, -400, 0}	{0, 0, 500}	$\mathbf{P}_E$ is singular
e2	h1 & g4 RMDPL	628	0.499061	0.75 (6.63)	{400,0,300 }	{ 0, 400, 300}	3D to 2D
e3	h1 & g4 Bres-3D	628	0.499061	0.28 (6.18)	{400,0,300 }	{ 0, 400, 300}	3D to 2D

The MaxErr of all the IPOs was smaller than the upper bound $\frac{\sqrt{2}}{2}$.

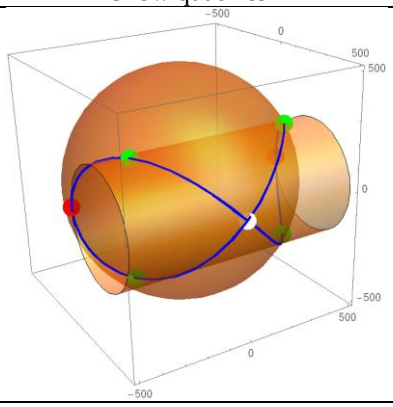
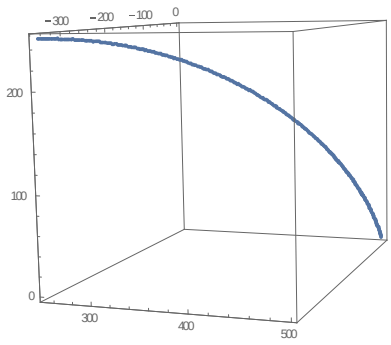
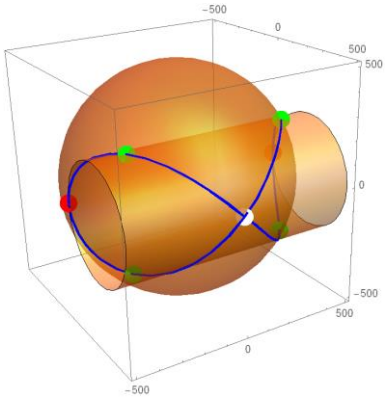
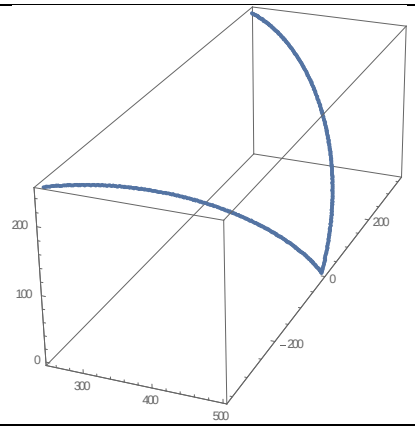
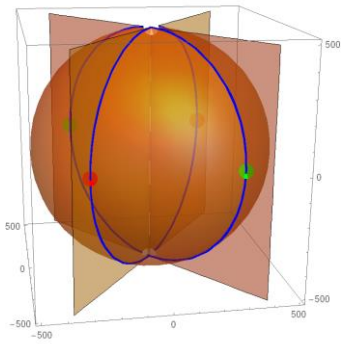
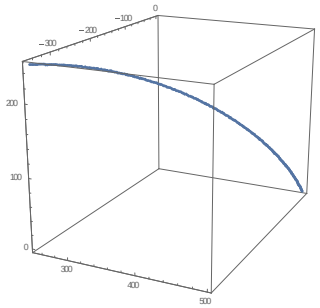
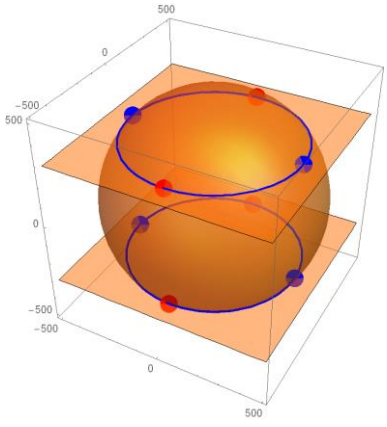
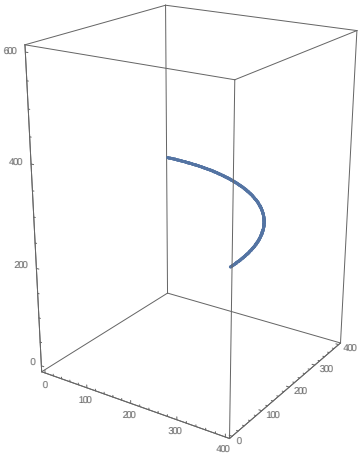
The error dErr[$\mathbf{P}_n$] of example e, was calculated with the 2D-distance function

$N[Abs[\sqrt{x_n^2 + y_n^2} - r]]$ instead of Mathematica's RegionDistance function.

The real advantage of Bresenham's 3D-curve algorithm is the extremely small execution time.

The imperfect Bresenham-3D-curve IPO becomes perfect when it deteriorates to 2D

Table 8 3D-curves of the examples b, c, d and e

Example	Show quadrics	QSIC
b1, b2, b3 $x^2+y^2+z^2-R^2 = 0$ $x^2 - R\,x + z^2 = 0$ $R = 500$		
b4, b5 $x^2+y^2+z^2-R^2 = 0$ $x^2 - R\,x + z^2 = 0$ $R = 500$		
d1, d2, d3 $x^2+y^2+z^2-R^2 = 0$ $16\,x^2 - 9\,y^2 = 0$ $R = 500$		
e2, e3 $x^2+y^2+z^2-R^2 = 0$ $z^2 - zo^2 = 0$ $R=500$ $r=400$ $z0 = \sqrt{R^2 - r^2} = 300$		

3 The perfect 3D-line IPO

The perfect 3D-line IPO, in CDF-form, can be found in [35].
The equation of the QSIC (21) simplifies to $\mathbf{f}[\mathbf{P}] = \mathbf{L} \times (\mathbf{P} - \mathbf{P}_S)$ equivalent with the symmetric function $\frac{x - x_S}{L_i} = \frac{y - y_S}{L_j} = \frac{z - z_S}{L_k}$ with $\mathbf{L} = \{L_i, L_j, L_k\}$ or $\mathbf{P} = \mathbf{P}_S + \mathbf{L} * t$. (Eq 68)
The perfect 3D-line IPO uses the RMDPL algorithm ($ 2.5) and the decision function (44) simplifies to $d[\mathbf{P}_u, \mathbf{P}_v] \equiv (\mathbf{L} \times (\mathbf{P}_u - \mathbf{P}_v)) \bullet \left(\mathbf{L} \times (\frac{\mathbf{P}_u + \mathbf{P}_v}{2} - \mathbf{P}_S) \right)$. (Eq 69)
The monotonic direction simplifies to $S_n = \{S_x, S_y, S_z\} = \text{Sign}[L_i, L_j, L_k]$ and the primary OoC-conditions (17; 5) simplify to $\mathbf{LA} \equiv \text{Abs}[\mathbf{L}] = \{LI, LJ, LK\} = S_n * \mathbf{L}$. (Eq 70)
The startpoint $\mathbf{P}_S$ has only influence on the initial conditions, and as the line is linear, the startpoint can be zero.
The decision variable $d[\mathbf{R}_u, \mathbf{R}_v]$ is a linear function with constant coefficients:
The perfect 3D-line algorithm computes the 21 listofPairs (52; 53) with $\mathbf{P}_A \equiv \mathbf{P}_n = \{ x_n, y_n, z_n \}$ and (50) and (69) with $\mathbf{L} = S_n * \mathbf{LA} = \{S_x * LI, S_y * LJ, S_z * LK\}$ using Mathematica's function
$d_{TE}[\mathbf{P}_u\_, \mathbf{P}_v\_] := \text{TensorExpand}[d[\mathbf{P}_u, \mathbf{P}_v] /. \{S_x^2 \to 1, S_y^2 \to 1, S_z^2 \to 1\}$, (Eq 71)
$\text{Duv} = d_{TE}[\mathbf{P}_u, \mathbf{P}_v] /. \{x_n \to 0, y_n \to 0, z_n \to 0\}$, (Eq 72)
$\text{duvx} = d_{TE}[\mathbf{P}_u, \mathbf{P}_v] - \text{Duv} /. \{S_x \to 1, x_n \to 1, y_n \to 0, z_n \to 0\}$, (Eq 73)
$\text{duvy} = d_{TE}[\mathbf{P}_u, \mathbf{P}_v] - \text{Duv} /. \{S_y \to 1, x_n \to 0, y_n \to 1, z_n \to 0\}$, (Eq 74)
$\text{duvz} = d_{TE}[\mathbf{P}_u, \mathbf{P}_v] - \text{Duv} /. \{S_z \to 1, x_n \to 0, y_n \to 0, z_n \to 1\}$. (Eq 75)
The symbolic decision variables (72 - 75) of all the pairwise points (52) are stored in respectively
listDV0 = (Eq 76)
{LI^2-2 LI LJ-2 LJ LK+LK^2,-2 LI LJ+LJ^2-2 LI LK+LK^2,LI^2+LJ^2-2 LI LK-2 LJ LK,LI^2-2 LI LJ+LJ^2-2 LI LK-2 LJ LK+2 LK^2,LI^2-2 LI LJ+2 LJ^2-2 LI LK-2 LJ LK+LK^2,2 LI^2-2 LI LJ+LJ^2-2 LI LK-2 LJ LK+LK^2,-LI^2+LJ^2-2 LI LK+2 LJ LK,2 LI LJ+LJ^2-2 LI LK-LK^2,LJ^2-2 LI LK+LK^2,2 LJ^2-2 LI LK,LI^2+LJ^2-2 LI LK,LI^2+2 LI LJ-2 LJ LK-LK^2,LI^2-2 LJ LK+LK^2,LI^2+LJ^2-2 LJ LK,2 LI^2-2 LJ LK,-2 LI LJ+2 LK^2,-2 LI LJ+LJ^2+LK^2,LI^2-2 LI LJ+LK^2,LJ^2-LK^2,LI^2-LK^2,LI^2-LJ^2};
listDVx = (Eq 77)
{-2 LI LJ,2 (LJ^2+LK^2),-2 LI LK,-2 (LI LJ-LJ^2-LK^2),2 (LJ^2+LK (-LI+LK)),-2 LI (LJ+LK),2 (LI LJ+LJ^2+LK^2),2 LI (LJ-LK),2 (LJ^2+LK^2),2 (LI LJ+LJ^2-LI LK+LK^2),-2 LI LK,-2 (LJ^2+LK (LI+LK)),-2 LI LJ,-2 LI LK,-2 (LI LJ+LJ^2+LI LK+LK^2),-2 (LI LJ-LJ^2-LI LK-LK^2),2 (LJ^2+LK^2),-2 LI LJ,2 LI (LJ-LK),-2 (LJ^2+LK (LI+LK)),-2 (LI LJ+LJ^2+LK^2)};
listDVy = (Eq 78)
{2 (LI^2+LK^2),-2 LI LJ,-2 LJ LK,2 (LI^2-LI LJ+LK^2),-2 LJ (LI+LK),2 (LI^2+LK (-LJ+LK)),-2 (LI^2+LI LJ+LK^2),-2 (LI^2+LK (LJ+LK)),-2 LI LJ,-2 (LI^2+LI LJ+LK (LJ+LK)),-2 LJ LK,2 LJ (LI-LK),2 (LI^2+LK^2),-2 LJ LK,2 (LI^2+LI LJ+LK (-LJ+LK)),2 (LI^2-LI LJ+LK (LJ+LK)),-2 LI LJ,2 (LI^2+LK^2),-2 (LI^2+LK (LJ+LK)),2 LJ (LI-LK),2 (LI^2+LI LJ+LK^2)};
listDVz = (Eq 79)
{-2 LJ LK,-2 LI LK,2 (LI^2+LJ^2),-2 (LI+LJ) LK,2 (LI^2+LJ^2-LI LK),2 (LI^2+LJ (LJ-LK)),-2 (LI-LJ) LK,2 (LI^2+LJ (LJ+LK)),-2 LI LK,2 (LI^2-LI LK+LJ (LJ+LK)),2 (LI^2+LJ^2),2 (LI^2+LJ^2+LI LK),-2 LJ LK,2 (LI^2+LJ^2),2 (LI^2+LJ (LJ-LK)+LI LK),-2 (LI^2+LI LK+LJ (LJ+LK)),-2 LI LK,-2 LJ LK,2 (LI^2+LJ (LJ+LK)),2 (LI^2+LJ^2+LI LK),2 (LI-LJ) LK};

The algorithm initializes the integers {d[u,v] } from listDV0; the results of the selection algorithm (51; 76) and the simplified state diagram {**P**min1, **P**min2, b02, bmajA1,bmajA2 } go to the feedrate algorithm PRG-cs and the IPO.
Some details of the perfect 3D-line IPO and its incremental updates:
To calculate MaxErr and the Cost, one needs dErr[P_]:=N[Abs[$\frac{\text{Norm}[L \times P]}{\text{Norm}[L]}$]] (67) which is less time-consuming than Mathematica's more general distance function (34 – 36).
More info about the algorithm:

a. As $\mathbf{P}_S \equiv \{0, 0, 0\}$, one initializes with { d[n7, n6], …, d[3,1] } ≡ listDVn = listDV0 (76). (Eq 80).
To speed up the selection of the best point, the candidate points { $\mathbf{P}_7$, $\mathbf{P}_6$, $\mathbf{P}_4$, $\mathbf{P}_2$, $\mathbf{P}_5$, $\mathbf{P}_3$, $\mathbf{P}_1$ } are associated with the integers nu ∈ { n7, n6, n4, n2, n5, n3, n1 } and multiplying (69) with S_n gives (72 – 79). This also transforms the 3D-line, internally, to the line in the 1st quadrant.

b. One starts with the selection algorithm with listDV1=listDVn. (Eq 81)
Apply (51) and replace $\mathbf{P}_u \rightarrow nu$, $\mathbf{P}_A \rightarrow \mathbf{P}_S \rightarrow \{0,0,0\}$ and $\mathbf{P}_{min1} \rightarrow nMin1$.
The computed nMin1 ≡ nu determines the increments $S\Delta 1 \equiv \{\Delta_x, \Delta_y, \Delta_z\}$ and each component of the increments equals 0 or 1.
The allowed moves are $\{\Delta_x, \Delta_y, \Delta_z\} * \{\delta_x, \delta_y, \delta_z\}$.

c. The update of listDV1 is listDV2 = listDV1+ $listDVx * \Delta_x + listDVy * \Delta_y + listDVz * \Delta_z$. (Eq 82)
The updates has no multiplication and involves only the additions of integer constants.

d. The best point is **P**min1 = $\mathbf{P}_A + \{S_x, S_y, S_z\} * \{\Delta_x, \Delta_y, \Delta_z\} * \{\delta_x, \delta_y, \delta_z\}$. (Eq 83)

e. Determine the associated major-axis parameter bmajA1 with $\mathbf{P}_{min1}$ as
bmajA1 = $\{\Delta x, \Delta y, \Delta z\}[[majN]] = S\Delta 1[[majN]]$ (Eq 84), hence bmajA1 equals 0 or 1.

f. When **P**min1 is no major axis move (bmajA1≡0), one computes **P**min2, nMin2, bmajA2 and the new increments $S\Delta 2 \equiv \{\Delta_x, \Delta_y, \Delta_z\}$ such that
Pmin2 = **P**min1 + $\{S_x, S_y, S_z\} * \{\Delta_x, \Delta_y, \Delta_z\} * \{\delta_x, \delta_y, \delta_z\}$. (Eq 85)
bmajA2 = $\{\Delta x, \Delta y, \Delta z\}[[majN]] = S\Delta 2[[majN]]$ (Eq 86), hence bmajA1 equals 0 or 1.

g. The update of listDV2 is listDV3 = listDV2+ $listDVx * \Delta_x + listDVy * \Delta_y + listDVz * \Delta_z$. (Eq 87)
Put the latest update equal to listDVn.

From (80), ((81), (82), (83), (85) and (87), the decision function listDVn corresponding with $\mathbf{P}_n = \{x_n, y_n, z_n\}$ is listDVn = listDV0+ $listDVx * x_n + listDVy * y_n + listDVz * z_n$. (Eq 88)

4 The perfect RMDPL 3D-curve IPO

The perfect 3D-curve IPO, without updates in CDF-form, can be found in [36].

4.1 Preliminary calculations of RMDPL and Bresenham's 3D-curve algorithms

Define the quadrics (7), with R = 500, r = 400, z0 = 300 (§ 2.9):

- $f1[x,y,z] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} A_1 & D_1 & F_1 & I_1 \\ D_1 & B_1 & E_1 & J_1 \\ F_1 & E_1 & C_1 & K_1 \\ I_1 & J_1 & K_1 & M_1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$,

  {A1, B1, C1, D1, E1, F1, I1, J1, K1, M1} = {1, 1, 1, 0, 0, 0, 0, 0, 0, $-R^2$},

  h4 = h3 = h2 = h1 = $f_1[x,y,z] = x^2 + y^2 + z^2 - R^2 = 0$,

- $f2[x,y,z] = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \cdot \begin{bmatrix} A_2 & D_2 & F_2 & I_2 \\ D_2 & B_2 & E_2 & J_2 \\ F_2 & E_2 & C_2 & K_2 \\ I_2 & J_2 & K_2 & M_2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$,

  $g1 = x^2 + z^2 - R\,x = 0$,

  $\Leftrightarrow$ {A2, B2, C2, D2, E2, F2, I2, J2, K2, M2} = $\{1, 0, 1, 0, 0, 0, -\frac{R}{2}, 0, 0, 0\}$,

  $g2 = 4 * x - 3 * y = 0$,

  $\Leftrightarrow$ {A2, B2, C2, D2, E2, F2, I2, J2, K2, M2} = $\{0, 0, 0, 0, 0, 0, 2, -\frac{3}{2}, 0, 0\}$,

g3 = 16* x² - 9* y² = 0,
⇔{A2, B2, C2, D2, E2, F2, I2, J2, K2, M2} = {16, -9, 0, 0, 0, 0,0, 0, 0, 0},
g4 = z² - $z_0$² = 0,
⇔{A2, B2, C2, D2, E2, F2, I2, J2, K2, M2} = {0, 0, 1, 0, 0, 0,0, 0, 0, -$z_0$²},
Calculate offline the extreme points Table 6 from (§ 2.6, (54), (55), (56)),

- Singular points PtsWHITE,
- Extreme non-singular points PtsBLUE,
- Extreme non-singular points PtsRED,
- Extreme non-singular points PtsGREEN.

From the extreme points and the four examples define respectively the rounded startpoint $\mathbf{P}_S$, endpoint $\mathbf{P}_E$, the possible moves Sδ = {1, 1, 1} (4)
and direction vector S_n = {Sx, Sy, Sz} = Sign[$\mathbf{P}_E$ -$\mathbf{P}_S$] (5; 57),

- Example b: $\mathbf{P}_S$ = {250, -354, 250}, $\mathbf{P}_E$ = {500, 000, 000}, Sn = {Sx, Sy, Sz} = { 1, 1, -1},
  b4 & b5: $\mathbf{P}_{S2}$ = {500, 1, 1}, $\mathbf{P}_{E2}$ = {250, 354, 250}, Sn = {Sx, Sy, Sz} = {-1, 1, 1},
- Example c: $\mathbf{P}_S$ = {300, -400, 0}, $\mathbf{P}_E$ = {0, 0, 500}, Sn = {Sx, Sy, Sz} = {-1, 1, 1},
- Example d: $\mathbf{P}_S$ = {300, -400, 0}, $\mathbf{P}_E$ = {0, 0, 500}, Sn = {Sx, Sy, Sz} = {-1, 1, 1},
- Example e: $\mathbf{P}_S$ = {400, 0, 300}, $\mathbf{P}_E$ = {0, 400, 300}, Sn = {Sx, Sy, Sz} = {-1, 1, 0},
  hence, δz = 0.

Calculate respectively X_1, Y_1, Z_1, W_1, $\mathbf{G}_1$, X_2, Y_2, Z_2, W_2, $\mathbf{G}_2$, from (10), (11), (12), (13), and (9).
Calculate the sign-factor (14) $S_L = \mathrm{Sign}[(\mathbf{G}_1[\mathbf{P}_s]\times\mathbf{G}_2[\mathbf{P}_s])\bullet\{S_x, S_y, S_z\}]$ from the startpoint and the direction vector S_n.
Define the startpoint $\mathbf{P}_A$ = $\mathbf{P}_S$ of the candidate points.
Calculate the vectors $\mathbf{T}_A$ (15), $\mathbf{W}_A$ (18) and its components in the startpoint $\mathbf{P}_A$.
Calculate the normalized length (61) LNcs = $2*\mathrm{Ceiling}[\ \mathrm{Norm}[\ \mathbf{P}_{monoE} - \mathbf{P}_{monoS}\ ] + .05\ ]$. (Eq 89)
Calculate the accumulator of the PRM-cs ACcs = $\frac{\mathrm{LNcs}}{2}$. (Eq 90)

4.2 Online calculations of RMDPL algorithms (While loop)

The practical decision function of the RMDPL-IPO is (44), with (50) and $\mathbf{P}_m = \frac{\mathbf{P}_u + \mathbf{P}_v}{2}$

$$d[\mathbf{P}_u, \mathbf{P}_v] \equiv (\mathbf{T}[\frac{\mathbf{P}_u + \mathbf{P}_v}{2}]\times(\mathbf{P}_u - \mathbf{P}_v))\bullet(\mathbf{T}[\frac{\mathbf{P}_u + \mathbf{P}_v}{2}]\times(\frac{\mathbf{P}_u + \mathbf{P}_v}{2}) + \mathbf{W}[\frac{\mathbf{P}_u + \mathbf{P}_v}{2}]).$$

The two terms of the dot product of (44) are $\mathbf{T}[\mathbf{P}_m]\times(\mathbf{P}_u - \mathbf{P}_v)$ and $\mathbf{T}[\mathbf{P}_m]\times\mathbf{P}_m + \mathbf{W}[\mathbf{P}_m]$.
The components $\{T_i[\mathbf{P}_A], T_j[\mathbf{P}_A], T_k[\mathbf{P}_A]\}$ of the direction vector $\mathbf{T}[\mathbf{P}_A]$ are quadrics in $\{x_A, y_A, z_A\}$ with known coefficients. These quadrics are fast and incremental updated using (22), the 2D-update [5, Item d. UpdXMYMFM, page 5] and the3D-update (§5.3). The updates from $\mathbf{T}[\mathbf{P}_A]$ to $\mathbf{T}[\mathbf{P}_m]$ are also fast and although the number of possible updates equals 21 (53), only 6 $d[\mathbf{P}_u, \mathbf{P}_v]$ updates and comparisons must be performed (51).
The term $\mathbf{T}[\mathbf{P}_m]\times(\mathbf{P}_u - \mathbf{P}_v)$ (Eq 91) is obtained without multiplication.
The components of $\mathbf{T}[\mathbf{P}_m]\times\mathbf{P}_m + \mathbf{W}[\mathbf{P}_m]$ are of the third order in $\{x_m, y_m, z_m\}$. The components of $\mathbf{T}[\mathbf{P}_m]$ and $\mathbf{W}[\mathbf{P}_m]$ are quadrics and pose no problems, but the term $\mathbf{T}[\mathbf{P}_m]\times\mathbf{P}_m$ = $\{z_m * T_j[\mathbf{P}_m] - y_m * T_k[\mathbf{P}_m],\ x_m * T_k[\mathbf{P}_m] - z_m * T_i[\mathbf{P}_m],\ y_m * T_i[\mathbf{P}_m] - x_m * T_k[\mathbf{P}_m]\}$ (Eq 92) needs some multiplications.
All perfect or imperfect algorithms need $\mathbf{T}[\mathbf{P}_A]$, therefore the preferred IPOs use (44) instead of (47). Equation (47) clearly shows the needed updates and the multiplications: the quadrics and their differences pose no incremental problems (compare these terms with the 2D-case). The terms $\{\mathbf{G}_{1m}\bullet\mathbf{G}_{2m}, \mathbf{G}_{1m}^2, \mathbf{G}_{2m}^2\}$ are all quadrics and the coefficients of these quadrics can be pre-computed. Hence, all the terms can be fast updated, but some multiplications are needed to compute (47).

The form of (44) for the 3D-line is $\left(\mathbf{T}[\mathbf{P}_m]\times(\mathbf{P}_u-\mathbf{P}_v)\right)\bullet\left(\mathbf{T}[\mathbf{P}_m]\times\mathbf{P}_m+\mathbf{W}[\mathbf{P}_m]\right)\equiv$ $\left(\mathbf{L}\times(\mathbf{P}_u-\mathbf{P}_v)\right)\bullet\left(\mathbf{L}\times(\mathbf{P}_m-\mathbf{P}_S)\right)$ and as the components of { $\mathbf{L}$, $\mathbf{P}_S$ } are constants, the 3D-line IPO is completely incremental (§ 3).
A 16-year-old paper used (47). The extreme points (§2.6) were not precalculated, and the algorithm had to find AI-solutions when arriving at an unknown extreme point. All the f_{rm}'s, G^2_{rm} 's and $\mathbf{G}_{1m}\bullet\mathbf{G}_{2m}$'s were quadrics and were incremental computed (22). The paper was never published, because it is impossible to publish it in a clear way, but the execution time was acceptable and some results for 6- and 26-connected QSICS are given (Appendix 7.2,Table 12, 6-connected QSICs are not shown).
The object of the paper is to represent the 3D IPOs as clearly as possible, therefore the while loops of the RMDPL-IPO and Bresenham' 3D-IPO will not be presented in incremental form.
Recompute Sδ={δx, δy, δz} (96) before starting the While-loop.

While loop of the RMDPL-IPO:

The while loop starts with the current best point $\mathbf{P}_A$ and set b02=bmajA1=bmajA2=False.
Update to $\mathbf{T}_A$=$\mathbf{T}$[$\mathbf{P}_A$] (15 or 22).
Compute the parameters of the PRM-cs algorithm (Eq 93):
TAM=Max[Abs[$\mathbf{T}_A$], TAN=Norm[$\mathbf{T}_A$]] and LMcs = $\text{Floor}\left[\frac{1}{\text{kcs}}\frac{\text{TAM}}{\text{TAN}}*\text{LNcs}+\frac{1}{2}\right]$, (Eq 94)
Update (18 or 18) to $\mathbf{W}_A$=$\mathbf{W}$[$\mathbf{P}_A$] . (Eq 95)
Update from {$\mathbf{T}_A$, $\mathbf{W}_A$} to {$\mathbf{T}$m, $\mathbf{W}_m$} with $\mathbf{P}_m=\frac{\mathbf{P}_u+\mathbf{P}_v}{2}$ using the candidate points (50) and depending on the value of $\mathbf{P}_u$ and $\mathbf{P}_v$ do the d[$\mathbf{P}_u$, $\mathbf{P}_v$]-update, and do (51).
The best selected point is $\mathbf{P}_{min1}$ = $\mathbf{P}_{A1}$ = $\mathbf{P}_u$.
Update to $\mathbf{T}_{A1}$=$\mathbf{T}$[$\mathbf{P}_{A1}$] (15 or 22) .
Compute the parameters of the PRM-cs algorithm:
TAM1=Max[Abs[$\mathbf{T}_{A1}$], TAN1=Norm[$\mathbf{T}_{A1}$]] and LMcs1 = $\text{Floor}\left[\frac{1}{\text{kcs}}\frac{\text{TAM1}}{\text{TAN1}}*\text{LNcs}+\frac{1}{2}\right]$.
The major axis move of $\mathbf{P}_A$ to $\mathbf{P}_{A1}$ or $\mathbf{P}_A$ to $\mathbf{P}_{A2}$ uses LMcs and LMcs1 is only used for the move of $\mathbf{P}_{A1}$ to $\mathbf{P}_{A2}$, but the examples b, c, d, and e could even then use LMcs.
Compute the major-axis index majA1 corresponding with the polar line vector TAM1:
majA1 = Switch[TAM1,Abs[TA1[[1]], 1, Abs[TA1[[2]], 2, Abs[TA1[[3]], 3].
Determine if the move of $\mathbf{P}_A$ to $\mathbf{P}_{A1}$ is a major axis move .
bmajA1 = Switch[majA1, 1, $x_{A1}\neq x_A$, 2, $y_{A1}\neq y_A$, 3, $z_{A1}\neq z_A$].
If bmajA1 is True,

- $\mathbf{P}_A$ to $\mathbf{P}_{A1}$ is a major axis move.
- Run the constant feedrate algorithm PRM-cs
  Label[StartCS1]; npuls = npuls+1; ACcs = ACcs – LMcs; If[ACcs > 0, Goto[StartCS1];
  ACcs = ACcs + LNcs; nkey = nkey.
- Update $\mathbf{P}_A$ and register the moves,
  $\mathbf{P}_A$ = {x_A, y_A, z_A} = $\mathbf{P}_{A1}$.
  listxyz26=Join[listxyz26,{ $\mathbf{P}_A$ }]. listInfo=Join[listInfo, {$\mathbf{P}_A$, npuls, nkey}].
  Or to determine MaxErr: dErrA=dErr[PA].listErr=Join[listErr[,{dErrA}].
- Recompute Sδ={δx, δy, δz}, (Eq 96)
  If[x_A == x_E, δx=0];
  If[y_A == y_E, δy=0];
  If[z_A == z_E, δz=0].

When $\mathbf{P}_{A1}$ is a major axis move (bmajA1=True) the IPO can be restarted, else the next point $\mathbf{P}_{A2}$ is computed.
]; (* End of bmajA1 is True *)

If bmajA1 is False, set bmajA2 equal to True and compute $\mathbf{P}_{A2}$ and b02.
Update to $\mathbf{W}_{A1}$=$\mathbf{W}$[$\mathbf{P}_{A1}$] (18 or 22).

Update from {$\mathbf{T}_{A1}$. $\mathbf{W}_{A1}$} to {**T**m, $\mathbf{W}_m$} with $\mathbf{P}_m = \frac{\mathbf{P}_u + \mathbf{P}_v}{2}$ using the candidate points (50) with $\mathbf{P}_A$ replaced by **P**A1 and depending on the value of $\mathbf{P}_u$ and $\mathbf{P}_v$ do the d[$\mathbf{P}_u$, $\mathbf{P}_v$]-update, and do (51).
The best selected point is $\mathbf{P}_{min2} = \mathbf{P}_{A2} = \mathbf{P}_u$.
Update to $\mathbf{T}_{A2}$=**T**[$\mathbf{P}_{A2}$] (15 or 22)and $\mathbf{W}_{A2}$=**W**[$\mathbf{P}_{A2}$] (18 or 22),
Do not compute the parameters of the PRM-cs algorithm.
Do not to recompute LMcs for the moves $\mathbf{P}_A$ to $\mathbf{P}_{A1}$ and $\mathbf{P}_A$ to $\mathbf{P}_{A2}$,
(use the values corresponding with $\mathbf{P}_A$), but recalculate LMcs for the move $\mathbf{P}_{A1}$ to $\mathbf{P}_{A2}$
Check if the Boolean b02 is True when $\mathbf{P}_{A2}$ is 26-connected with $\mathbf{P}_A$:
b02 = MemberQ[{0,0,0}, {0,1,0}, {0,1,1}, {1,0,0}, {1,0,1}, {1,1,0}, {1,1,1},Abs[$\mathbf{P}_{A2}$-$\mathbf{P}_A$]].
If[Not[b02],

- $\mathbf{P}_A$ to $\mathbf{P}_{A1}$ is a non-major axis move, and $\mathbf{P}_{A1}$ to $\mathbf{P}_{A2}$ is a major axis move.
- Update $\mathbf{P}_A$ and register the moves:
  $\mathbf{P}_A = \{x_A, y_A, z_A\} = \mathbf{P}_{A1}$.
  listxyz26=Join[listxyz26,{ $\mathbf{P}_A$ }]; listInfo=Join[listInfo, {$\mathbf{P}_A$, npuls, nkey}];
  And to determine MaxErr: dErrA=dErr[PA];listErr=Join[listErr[,{dErrA}].
- Recompute Sδ (96), If[$x_A == x_E$, δx=0]; If[$y_A == y_E$, δy=0]; If[$z_A == z_E$, δz=0].
- Run the constant feedrate algorithm PRM-cs for the move $\mathbf{P}_{A1}$ to $\mathbf{P}_{A2}$ using LMcs1:
  Label[StartCS2]; npuls = npuls+1; ACcs = ACcs – LMcs1; If[ACcs > 0, Goto[StartCS2];
  ACcs = ACcs + LNcs; nkey = nkey.
- Update $\mathbf{P}_A$ and register the moves:
  $\mathbf{P}_A = \{x_A, y_A, z_A\} = \mathbf{P}_{A2}$.
  listxyz26=Join[listxyz26,{ $\mathbf{P}_A$ }]; listInfo=Join[listInfo, {$\mathbf{P}_A$, npuls, nkey}].
  And to determine MaxErr: dErrA=dErr[PA];listErr=Join[listErr[,{dErrA}].
- Recompute Sδ (96), If[$x_A == x_E$, δx=0]; If[$y_A == y_E$, δy=0]; If[$z_A == z_E$, δz=0].

]; (* End of Not[b02] *)
If[b02,

- $\mathbf{P}_A$ to $\mathbf{P}_{A2}$ is a 26-connected major axis move.
- Run the constant feedrate algorithm PRM-cs:
  Label[StartCS2]; npuls = npuls+1; ACcs = ACcs – LMcs; If[ACcs > 0, Goto[StartCS2];
  ACcs = ACcs + LNcs; nkey = nkey.
- Update $\mathbf{P}_A$ and register the moves:
  $\mathbf{P}_A = \{x_A, y_A, z_A\} = \mathbf{P}_{A2}$;
  listxyz26=Join[listxyz26,{ $\mathbf{P}_A$ }]; listInfo=Join[listInfo, {$\mathbf{P}_A$, npuls, nkey}].
  And to determine MaxErr: dErrA=dErr[PA];listErr=Join[listErr[,{dErrA}].
- Recompute Sδ (96), If[$x_A == x_E$, δx=0]; If[$y_A == y_E$, δy=0]; If[$z_A == z_E$, δz=0].

]; (* End of b02 *)
]; (* End of bmajA1 is False *)
]; (* End of While loop *)

5 Bresenham's 3D-curve IPO

Bresenham's 3D-curve IPO, without updates in CDF-form, can be found in [37].
Bresenham's 3D algorithm is imperfect because

1. Each move is a major-axis move, but that simplifies the constant feedrate calculation, and instead of 21 decision variables (53), only 2 decision variables per major-axis direction are needed, therefore its speed is significant higher;
2. The decision function is simpler but does not measure the 3D distance to the polar line of the curve. Therefore, the accuracy is lower, and MaxErr is not minimal, but as a bonus, the MaxErr keeps bounded to $\frac{\sqrt{2}}{2} \simeq 0.707$.
3. It does not apply the simplified state diagram.
   Each major axis move runs the feedrate algorithm PRM-cs.

Each item {1, 2, 3} lowers the accuracy and increases the execution speed, therefore Bresenham's curve IPO is extremely fast and remains within the accuracy bound.

Bresenham's 3D-curve algorithm degenerates to Bresenham's circle algorithm when you apply the intersection of a sphere and a plane perpendicular to the z-axis, but in that case, you get the perfect 2D circle algorithm, the MaxErr is bounded to 0.5 and is minimal. As you may expect, Bresenham's 3D-curve IPO degenerates to Bresenham's 3D-line. These properties are not special because it occurs with the 3D-RMDPL-IPOs too. This means, as Bresenham's 3D-curve IPO is a subclass of the 3D-RMDPL-IPOs, that all reference pulse IPOs, basically, are RMDPL-IPOs.
What about the practical decision function (44) ?
The candidate point $\mathbf{P}_u$ and $\mathbf{P}_v$ are obtained from $\mathbf{P}_A$ and the monotonic direction Sn (5;50).
The dot product (44) can be written as $d[\mathbf{P}_u,\mathbf{P}_v] = \mathbf{f}_{uv} \bullet \mathbf{f}_m$ with $\mathbf{P}_m = \frac{\mathbf{P}_u + \mathbf{P}_v}{2}$,
$\mathbf{f}_{uv} = \mathbf{T}[\mathbf{P}_m] \times (\mathbf{P}_u - \mathbf{P}_v)$ and $\mathbf{f}_m = \mathbf{T}[\mathbf{P}_m] \times \mathbf{P}_m + \mathbf{W}[\mathbf{P}_m]$. The vector $\mathbf{f}_m[\mathbf{P}] \triangleq \mathbf{T}[\mathbf{P}_m] \times \mathbf{P} + \mathbf{W}[\mathbf{P}_m]$ is the polar line of the QSIC with respect to the point $\mathbf{P}_m$ and the residue $\mathbf{f}_m = \mathbf{f}_m[\mathbf{P}_m]$ in the point $\mathbf{P}_m$ is the residue of the QSIC in the point $\mathbf{P}_m$. In the same way $\mathbf{f}_{uv}$ equals $\mathbf{f}_{uv} = \mathbf{f}_m[\mathbf{P}_u] - \mathbf{f}_m[\mathbf{P}_v]$. The vector function $\mathbf{f}_m[\mathbf{P}]$ equals
$\{f_{mx}[\mathbf{P}], f_{my}[\mathbf{P}], f_{mz}[\mathbf{P}]\} = \{\mathbf{f}_m[\mathbf{P}][[1]], \mathbf{f}_m[\mathbf{P}][[2]], \mathbf{f}_m[\mathbf{P}][[3]]\}$ (Eq 97) and the components are the projections of $\mathbf{f}_m[\mathbf{P}]$ on respectively the yz-plane, the zx-plane, and the xy-plane. The dot product (44) equals $(f_{mx}[\mathbf{P}_u] - f_{mx}[\mathbf{P}_v]) * f_{mx}[\mathbf{P}_m] + (f_{my}[\mathbf{P}_u] - f_{my}[\mathbf{P}_v]) * f_{my}[\mathbf{P}_m] + (f_{mz}[\mathbf{P}_u] - f_{mz}[\mathbf{P}_v]) * f_{mz}[\mathbf{P}_m]$. (Eq 98)
Define $Pdx[\mathbf{P}_u,\mathbf{P}_v]$ as the product of the x-projections of $\mathbf{f}_{uv}$ and the x-projection of $\mathbf{f}_m$ and define Pdy and Pdz analogously. Hence,
$Pdx[\mathbf{P}_u,\mathbf{P}_v] \triangleq (f_{mx}[\mathbf{P}_u] - f_{mx}[\mathbf{P}_v]) * f_{mx}[\mathbf{P}_m]$, (Eq 99)
$Pdy[\mathbf{P}_u,\mathbf{P}_v] \triangleq (f_{my}[\mathbf{P}_u] - f_{my}[\mathbf{P}_v]) * f_{mz}[\mathbf{P}_m]$, (Eq 100)
$Pdz[\mathbf{P}_u,\mathbf{P}_v] \triangleq (f_{mz}[\mathbf{P}_u] - f_{mz}[\mathbf{P}_v]) * f_{mz}[\mathbf{P}_m]$. (Eq 101)
and (44) becomes $d[\mathbf{P}_u,\mathbf{P}_v] = Pdx[\mathbf{P}_u,\mathbf{P}_v] + Pdy[\mathbf{P}_u,\mathbf{P}_v] + Pdz[\mathbf{P}_u,\mathbf{P}_v]$. (Eq 102)
The scalar function $Pdx[\mathbf{P}_u,\mathbf{P}_v]$ measures the x-distance to the line $\{\mathbf{P}_u,\mathbf{P}_v\}$. (Eq 103)
The scalar function $Pdy[\mathbf{P}_u,\mathbf{P}_v]$ measures the y-distance to the line $\{\mathbf{P}_u,\mathbf{P}_v\}$. (Eq 104)
The scalar function $Pdz[\mathbf{P}_u,\mathbf{P}_v]$ measures the z-distance to the line $\{\mathbf{P}_u,\mathbf{P}_v\}$. (Eq 105)
Bresenham's 3D-curve algorithm is a major-axis algorithm, therefore three possible tripods are formed by the legs {{ {$\mathbf{P}_1$,$\mathbf{P}_A$}, {$\mathbf{P}_1$,$\mathbf{P}_2$}, {$\mathbf{P}_1$,$\mathbf{P}_6$}}, { {$\mathbf{P}_3$,$\mathbf{P}_A$}, {$\mathbf{P}_3$,$\mathbf{P}_2$}, {$\mathbf{P}_3$,$\mathbf{P}_4$}}, { {$\mathbf{P}_5$,$\mathbf{P}_A$}, {$\mathbf{P}_5$,$\mathbf{P}_4$}, {$\mathbf{P}_5$,$\mathbf{P}_6$}} }. The $\mathbf{P}_1$-tripod corresponds with the major x-axis and the minimal z-distance to the leg {$\mathbf{P}_1$,$\mathbf{P}_2$} determines the best point of {$\mathbf{P}_1$,$\mathbf{P}_2$}; the minimal y-distance to the leg {$\mathbf{P}_1$,$\mathbf{P}_6$} determines the best point of {$\mathbf{P}_1$,$\mathbf{P}_6$}. Hence Bresenham's decision functions of the $\mathbf{P}_1$-tripod are dBz[$\mathbf{P}_2$, $\mathbf{P}_1$]=If[Pdz[$\mathbf{P}_2$,$\mathbf{P}_1$] ≤ 0, $y_A = y_A + S_y$] and dBy[$\mathbf{P}_6$, $\mathbf{P}_1$]=If[Pdy[$\mathbf{P}_6$,$\mathbf{P}_1$] ≤ 0, $z_A = z_A + S_z$]. The Bresenham's decision function dBz[$\mathbf{P}2$,$\mathbf{P}_1$] does not only select $\mathbf{P}2$, but it selects the y-move, analogous for Bresenham's decision function dBy[$\mathbf{P}_6$,$\mathbf{P}_1$]. Therefore it also selects $\mathbf{P}_7$ when both moves are selected (Appendix 7.1).
When the distance equals zero, Bresenham's criterion applies the priority rules too.
So, the real candidate points of the $\mathbf{P}_1$-tripod are {$\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_6$, $\mathbf{P}_7$}, but Bresenham's 3D-curve algorithm needs only two legs and two comparison [Appendix 7.1].
Table 9 shows the Bresenham's decision functions for the three tripods.

Table 9 Bresenham's decision functions for the three tripod

Tripod	Decision function leg 1	Decision function leg 2
{ $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_6$ } $x_A = x_A + S_x$ $\mathbf{P}_2$-$\mathbf{P}_1$ ={0,S_y,0} $\mathbf{P}_6$-$\mathbf{P}_1$ ={0,0,S_z}	dBz[$\mathbf{P}_2$, $\mathbf{P}_1$] = If[Pdz[$\mathbf{P}_2$,$\mathbf{P}_1$] ≤ 0, $y_A = y_A + S_y$] $\mathbf{P}_2$ - $\mathbf{P}_1$ = { 0, S_y, 0}	dBy[$\mathbf{P}_6$, $\mathbf{P}_1$] = If[Pdy[$\mathbf{P}_6$,$\mathbf{P}_1$] ≤ 0, $z_A = z_A + S_z$] $\mathbf{P}_6$ - $\mathbf{P}_1$ = { 0, 0, S_z}

	$\mathrm{Det}\left[\begin{pmatrix} S_x & S_y \\ 0 & S_y \end{pmatrix}\right] \times f_{mz}\left[\frac{\mathbf{P}_2+\mathbf{P}_1}{2}\right]$ Xor[BSy, BSx, fmz[P1, P2] ≤ 0]	$-\mathrm{Det}\left[\begin{pmatrix} S_x & S_z \\ 0 & S_z \end{pmatrix}\right] \times f_{my}\left[\frac{\mathbf{P}_6+\mathbf{P}_1}{2}\right]$ Xor[BSz, BSx, fmy[P1, P6] ≥ 0]
{ P$_3$**, P**$_4$**, P**$_2$ **}** $y_A = y_A + S_y$ **P**$_4$**-P**$_3$ $=\{0,0,S_z\}$ **P**$_2$**-P**$_3$ $=\{S_x,0,0\}$	dBx[**P**$_4$, **P**$_3$] = If[Pdx[**P**$_4$,**P**$_3$] ≤ 0, $z_A = z_A + S_z$] **P**$_4$ **- P**$_3$ = { 0, 0, S_z} $\mathrm{Det}\left[\begin{pmatrix} S_y & S_z \\ 0 & S_z \end{pmatrix}\right] \times f_{mx}\left[\frac{\mathbf{P}_4+\mathbf{P}_3}{2}\right]$ Xor[BSz, BSy, fmx[P3, P4] ≤ 0]	dBz[**P**$_2$, **P**$_3$] = If[Pdz[**P**$_2$,**P**$_3$] ≤ 0, $x_A = x_A + S_x$] **P**$_2$ **- P**$_3$ = { S_x, 0, 0} $\mathrm{Det}\left[\begin{pmatrix} S_x & S_y \\ S_x & 0 \end{pmatrix}\right] \times f_{mz}\left[\frac{\mathbf{P}_2+\mathbf{P}_3}{2}\right]$ Xor[BSx, BSy, fmz[P3, P2] ≥ 0]
{ P$_5$**, P**$_6$**, P**$_4$ **}** $z_A = z_A + S_z$ **P**$_6$**-P**$_5$ $=\{S_x,0,0\}$ **P**$_4$**-P**$_5$ $=\{0,S_y,0\}$	dBy[**P**$_6$, **P**$_5$] = If[Pdy[**P**$_6$,**P**$_5$] ≤ 0, $x_A = x_A + S_x$] **P**$_6$ **- P**$_5$ = { S_x, 0, 0} $-\mathrm{Det}\left[\begin{pmatrix} S_x & S_z \\ S_x & 0 \end{pmatrix}\right] \times f_{my}\left[\frac{\mathbf{P}_6+\mathbf{P}_5}{2}\right]$ Xor[BSx, BSz, fmy[P5, P6] ≤ 0]	dBx[**P**$_4$, **P**$_5$] = If[Pdx[**P**$_4$,**P**$_5$] ≤ 0, $y_A = y_A + S_y$] **P**$_4$ **- P**$_5$ = { 0, S_y, 0} $\mathrm{Det}\left[\begin{pmatrix} S_y & S_z \\ S_y & 0 \end{pmatrix}\right] \times f_{mx}\left[\frac{\mathbf{P}_4+\mathbf{P}_5}{2}\right]$ Xor[BSy, BSz, fmx[P5, P4] ≥ 0]

5.1 Simplification of the first term of (99), (100) and (101):

The difference **P**$_2$ - **P**$_1$ has always two zero components.

The primary OoC condition [2, §3] demands that the direction of the vector **T**[**P**$_m$] equals the direction of the monotonic vector $S_n \equiv \{S_x, S_y, S_z\}$ (5; 17).

Hence, the primary conditions in 3D become

$$\{S_x \left|T_x\left[\mathbf{P}_m\right]\right|, S_y \left|T_y\left[\mathbf{P}_m\right]\right|, S_z \left|T_z\left[\mathbf{P}_m\right]\right|\} = \{T_x\left[\mathbf{P}_m\right], T_y\left[\mathbf{P}_m\right], T_z\left[\mathbf{P}_m\right]\} = \mathbf{T}\left[\mathbf{P}_m\right]. \text{ (Eq 106)}$$

The 1st term of (101) equals $(f_{mz}[\mathbf{P}_2] - f_{mz}[\mathbf{P}_1]) = \begin{pmatrix} 0 & 0 & 1 \\ T_x\left[\mathbf{P}_m\right] & T_y\left[\mathbf{P}_m\right] & 0 \\ 0 & S_y & 0 \end{pmatrix} = S_x * S_y * \left|T_x\left[\mathbf{P}_m\right]\right|$

As only the sign of the 1st term is important, the result can be written as $\mathrm{Det}\left[\begin{pmatrix} S_x & S_y \\ 0 & S_y \end{pmatrix}\right]$.

Bresenham's decision function for the leg **P**$_2$ **P**$_1$ becomes

$$\mathrm{dBz}[\mathbf{P}_2, \mathbf{P}_1] = \mathrm{Det}\left[\begin{pmatrix} S_x & S_y \\ 0 & S_y \end{pmatrix}\right] \times f_{mz}\left[\frac{\mathbf{P}_2+\mathbf{P}_1}{2}\right]. \text{ (Eq 107)}$$

The 1st term of (100) equals $(f_{my}[\mathbf{P}_6] - f_{my}[\mathbf{P}_1]) = \begin{pmatrix} 0 & 1 & 0 \\ T_x\left[\mathbf{P}_m\right] & 0 & T_z\left[\mathbf{P}_m\right] \\ 0 & 0 & S_z \end{pmatrix} = -S_x * S_z * \left|T_x\left[\mathbf{P}_m\right]\right|$

As only the sign of the 1st term is important, the result can be written as $-\mathrm{Det}\left[\begin{pmatrix} S_x & S_z \\ 0 & S_z \end{pmatrix}\right]$.

Bresenham's decision function for the leg **P**$_6$ **P**$_1$ becomes

$$\mathrm{dBy}[\mathbf{P}_6, \mathbf{P}_1] = -\mathrm{Det}\left[\begin{pmatrix} S_x & S_z \\ 0 & S_z \end{pmatrix}\right] \times f_{my}\left[\frac{\mathbf{P}_6+\mathbf{P}_1}{2}\right]. \text{ (Eq 108)}$$

Analogous:

$$\mathrm{dBx}[\mathbf{P}_4, \mathbf{P}_3] = \mathrm{Det}\left[\begin{pmatrix} S_y & S_z \\ 0 & S_z \end{pmatrix}\right] \times f_{mx}\left[\frac{\mathbf{P}_4+\mathbf{P}_3}{2}\right], \text{ (Eq 109)}$$

$$\mathrm{dBz}[\mathbf{P}_2, \mathbf{P}_3] = \mathrm{Det}\left[\begin{pmatrix} S_x & S_y \\ S_x & 0 \end{pmatrix}\right] \times f_{mz}\left[\frac{\mathbf{P}_2+\mathbf{P}_3}{2}\right], \text{ (Eq 110)}$$

$$dBy[\mathbf{P}_6, \mathbf{P}_5] = -Det\left[\begin{pmatrix} S_x & S_z \\ S_x & 0 \end{pmatrix}\right] \times f_{my}\left[\frac{\mathbf{P}_6 + \mathbf{P}_5}{2}\right], \text{ (Eq 111)}$$

$$dBx[\mathbf{P}_4, \mathbf{P}_5] = Det\left[\begin{pmatrix} S_y & S_z \\ S_y & 0 \end{pmatrix}\right] \times f_{mx}\left[\frac{\mathbf{P}_4 + \mathbf{P}_5}{2}\right]. \text{ (Eq 112)}$$

Table 9 shows the results of the decision functions of all the cases and the logical decision functions,

- Define the logical components {BSx, BSy, BSz} of the monotonic direction vector Sn (5; 57),
  BSx = TrueQ[Sx==1], (Eq 113)
  BSy = TrueQ[Sy==1], (Eq 114)
  BSz = TrueQ[Sz==1]. (Eq 115)
- Define the logical decision functions,
  BdBz[$\mathbf{P}_2$, $\mathbf{P}_1$] =Xor[BSy, BSx, fmz[P1, P2] ≤ 0], If [BdBz[$\mathbf{P}_2$, $\mathbf{P}_1$], yA = yA + Sy], (Eq 116)
  BdBy[$\mathbf{P}_6$, $\mathbf{P}_1$] =Xor[BSz, BSx, fmy[P1, P6] ≥ 0], If [BdBy[$\mathbf{P}_6$, $\mathbf{P}_1$], zA = zA + Sz], (Eq 117)
  BdBx[$\mathbf{P}_4$, $\mathbf{P}_3$] =Xor[BSz, BSy, fmx[P3, P4] ≤ 0], If [BdBx[$\mathbf{P}_4$, $\mathbf{P}_3$], zA = zA + Sz], (Eq 118)
  BdBz[$\mathbf{P}_2$, $\mathbf{P}_3$] =Xor[BSx, BSy, fmz[P3, P2] ≥ 0], If [BdBz[$\mathbf{P}_2$, $\mathbf{P}_3$], xA = xA + Sx], (Eq 119)
  BdBy[$\mathbf{P}_6$, $\mathbf{P}_5$] =Xor[BSx, BSz, fmy[P5, P6] ≤ 0], If [BdBy[$\mathbf{P}_6$, $\mathbf{P}_5$], xA = yA + Sx],(Eq 120)
  BdBx[$\mathbf{P}_4$, $\mathbf{P}_5$] =Xor[BSy, BSz, fmx[P5, P4] ≥ 0], If [BdBx[$\mathbf{P}_4$, $\mathbf{P}_5$], yA = yA + Sy]. (Eq 121)

5.2 While loop of Bresenham's 3D-curve-IPO:

Initialize the normalized length LNcs=2*Ceiling[Norm[PE-PS]+0.5] and recompute Sδ (96).
The While-loop of Bresenham's IPO is easier than the While-loop of the RMDPL-IPO.
The while loop starts with the current best point $\mathbf{P}_A$.
Use the polar line of the QSIC with respect to the startpoint $\mathbf{P}_A$.
Update to $\mathbf{T}_A$=$\mathbf{T}$[$\mathbf{P}_A$] (15 or 22).
Compute the parameters of the PRM-cs algorithm (Eq 122):

- TAM=Max[Abs[$\mathbf{T}_A$], TAN=Norm[$\mathbf{T}_A$]] and LMcs = $\text{Floor}\left[\frac{1}{\text{kcs}}\frac{\text{TAM}}{\text{TAN}} * \text{LNcs} + \frac{1}{2}\right]$, (Eq 123)
- Update (18 or 22) to $\mathbf{W}_A$=$\mathbf{W}$[$\mathbf{P}_A$] . (Eq 124)

Compute the major-axis index majA corresponding with the polar line vector TAM:

- majA = Switch[TAM, Abs[$\mathbf{T}_A$[[1]], 1, Abs[$\mathbf{T}_A$[[2]], 2, Abs[$\mathbf{T}_A$[[3]], 3].

If majA == 1 and a x-move is possible or δx == 1,

- Calculate the tripod point $\mathbf{P}_1$ = $\mathbf{P}_A$ +{S_x, 0, 0} and make a x-move .x_A = x_A +S_x, calculate the legs $\mathbf{P}_2$ and $\mathbf{P}_6$.
- If a y-move is possible, apply If[Xor[BSy, BSx, fmz[P1, P2] ≤ 0], y_A = y_A +S_y.
- If a z-move is possible, apply If[Xor[BSz, BSx, fmy[P1, P6] ≥ 0], z_A = z_A +S_z.

If majA == 2 and a y-move is possible or δy == 1,

- Calculate the tripod point $\mathbf{P}_3$ = $\mathbf{P}_A$ +{0, S_y, 0} and make a y-move .y_A = y_A +S_y, calculate the legs $\mathbf{P}_2$ and $\mathbf{P}_6$.
- If a z-move is possible, apply If[Xor[BSz, BSy, fmx[P3, P4] ≤ 0], z_A = z_A +S_z.
- If a x-move is possible, apply If[Xor[BSx, BSy, fmz[P3, P2] ≥ 0], x_A = x_A +S_x.

If majA == 3 and a z-move is possible or δz == 1,

- Calculate the tripod point $\mathbf{P}_5$ = $\mathbf{P}_A$ +{0, 0, S_z} and make a z-move .z_A = z_A +S_z, calculate the legs $\mathbf{P}_2$ and $\mathbf{P}_6$.
- If a x-move is possible, apply If[Xor[BSx, BSz, fmy[P5, P6] ≤ 0], x_A = x_A +S_x.
- If a y-move is possible, apply If[Xor[BSy, BSz, fmx[P5, P4] ≥ 0], y_A = y_A +S_y.

Update $\mathbf{P}_A = \{x_A, y_A, z_A\}$

- Run the constant feedrate algorithm PRM-cs
  Label[StartCS1]; npuls = npuls+1; ACcs = ACcs – LMcs; If[ACcs > 0,
  Goto[StartCS1]; ACcs = ACcs + LNcs; nkey = nkey.

- Register the moves:
  listxyz26=Join[listxyz26,{ $\mathbf{P}_A$ }]. listInfo=Join[listInfo, {$\mathbf{P}_A$, npuls, nkey}].
  Or to determine MaxErr: dErrA=dErr[PA].listErr=Join[listErr[,{dErrA}].
- Recompute Sδ (96), If[$x_A == x_E$, δx=0]; If[$y_A == y_E$, δy=0]; If[$z_A == z_E$, δz=0].

]; (* End of While loop *)

The object of the paper is to represent the 3D IPOs as clearly as possible, therefore the While loop of the Bresenham' 3D-IPO will not be presented in incremental form.

5.3 Possible but not implemented incremental updates:

As stated in § 4.2 and § 4.3, the components of $\mathbf{f}_m = \mathbf{T}[\mathbf{P}_m] \times \mathbf{P}_m + \mathbf{W}[\mathbf{P}_m]$ can be incremental updated. The object of the paper is to represent the 3D IPOs as clearly as possible, therefore the while loops of the Bresenham' 3D-curve-IPO will not be presented in incremental form.
The Bresenham-updates are easier than the RMDPL-updates, that can be seen from the first equation of (20) and the update equation (22). The updates of
$\mathbf{f}_m = S_L * (f_1[\mathbf{P}_m] * \mathbf{G}_2[\mathbf{P}_m] - f_2[\mathbf{P}_m] * \mathbf{G}_1[\mathbf{P}_m])$ from $\mathbf{P}_A$ to $\mathbf{P}_m = \mathbf{P}_A + \{S_x * \Delta_x, S_y * \Delta_y, S_z * \Delta_z\}$
for $r = \{1, 2\}$ (Eq 125), (10), (11), (12), and (13) become

- $X_{rm} = X_{rA} + S_x A_r \Delta_x + S_y D_r \Delta_y + S_z F_r \Delta_z$, (Eq 126)
  $Y_{rm} = Y_{rA} + S_x D_r \Delta_x + S_y B_r \Delta_y + S_z E_r \Delta_z$, (Eq 127)
  $Z_{rm} = Z_{rA} + S_x F_r \Delta_x + S_y E_r \Delta_y + S_z C_r \Delta_z$, (Eq 128)
  $W_{rm} = W_{rA} + S_x I_r \Delta_x + S_y J_r \Delta_y + S_z K_r \Delta_z$. (Eq 129)
  Hence, $\mathbf{G}_r[\mathbf{P}_m] = \{X_{rm}, Y_{rm}, Z_{rm}\}$. (Eq 130)
- The updates (22) equal
  $f_r[\mathbf{P}_m] = f_r[\mathbf{P}_A] + S_x \Delta_x (X_{rm} + X_{rA}) + S_y \Delta_y (Y_{rm} + Y_{rA}) + S_z \Delta_z (Z_{rm} + Z_{rA})$. (Eq 131)
- The updates to $\{\mathbf{G}_1[\mathbf{P}_m], \mathbf{G}_2[\mathbf{P}_m], f_1[\mathbf{P}_m], f_2[\mathbf{P}_m]\}$ are simple, but the updates to
  $\mathbf{f}_m = \{f_{mx}, f_{my}, f_{mz}\}$ demand two multiplications pro projection. (Eq 132)
- When the best point $\mathbf{P}_n$ (50) becomes the new $\mathbf{P}_A$ you must update from the old $\mathbf{P}_A$ to the best point $\mathbf{P}_n$, using the same update equations with $\mathbf{P}_m \equiv \mathbf{P}_n$. When $\mathbf{P}_n$ becomes the new $\mathbf{P}_A$ then
  $\mathbf{P}_A = \mathbf{P}_n$, $\{X_{rA}, Y_{rA}, Z_{rA}, W_{rA}\} = \{X_{rn}, Y_{rn}, Z_{rn}, W_{rn}\}$, $\mathbf{G}_r[\mathbf{P}_A] = \mathbf{G}_r[\mathbf{P}_n]$ and
  $f_r[\mathbf{P}_A] = f_r[\mathbf{P}_n]$. (Eq 133)

6 Conclusion

The paper presented three new 26-connected constant feedrate IPOs that can be used in practical situations in CNC machining tools. The RMDPL-IPOs generate perfect 3D-lines or curves with minimal MaxErr. A simple embedded microprocessor, for example the 32-bit PIC32, can easily and fast generate the interrupts and the computations in C or C++. The less accurate but super-fast Bresenham-3D-curve-IPO can be used in many practical situations as MaxErr is bounded to $\sqrt{2}/2$. Quadrics and QSICs are basic objects of CAD/CAM Systems.
The criterion of the Relative Minimal Distance of two pairwise candidate points to the Polar Line of the curve (QSIC) and cascading the ultra-fast PRM-cs to 26-connected curves are crucial. Theoretical, the RMDPL is fundamental, it is the core of all the successful 2D-incremental step algorithms and this paper proves that it is the core of the 3D incremental step algorithms or 3D reference IPOs too. Many papers (Bresenham 2D-line and circle algorithms, Van Aken's & Pitteway's midpoint algorithms, and all the incremental algorithms of the NATO ASI Series F books [33; 34] given to me by J. E. Bresenham) tried to explain the 2D-incremental step algorithms, the conflicts between the results of the midpoint and two-point methods (OoA errors). The sampling of curves was theorical clear and solved the constant feedrate problem when the arclength of the curve was known. Now, the theoretical background of 2D and 3D incremental-step curves is clear [2], the OoA- and the OoC-problems are solved and implementing constant

feedrate with high accuracy to a 26-connected curve turns out to be a piece of cake in contrast with the sampled-data curves.
CNC machining tools which use RMDPL will be simpler, faster, smoother, and more accurate than the current machining tools based on sampled-data systems.
All IPOs can be converted to constant feedrate listSIM-IPOs which can be used in real time in simplified rigid CNC machine tools [1;4, § 7].

6.1 Open problem

This paper completes my papers on incremental 8- and 26-connected curves. It does not mean that there are no unsolved or open problems : e.g. the determination of the polar line of NURBS with respect to a given point $\mathbf{P}_m$.
Nowadays the NURBS must be converted to composite Bézier curves, and finally the curves are converted to tangent conics or QSICS [1; 4, § 8 and §10.1], [32].

7 Appendix

7.1 RMDPL 2D

When you apply the RMDPL criterion (51) to the case $\{\delta x, \delta y, \delta z,\} \equiv \{1, 1, 0\}$ and when you replace the sign "≤" of the Booleans Bd23 = Boole[$\mathbf{P}_2, \mathbf{P}_3$] ≤ 0], Bd21 = Boole[$\mathbf{P}_2, \mathbf{P}_1$] ≤ 0], Bd31 = Boole[$\mathbf{P}_3, \mathbf{P}_1$] ≤ 0 by the signs "<" and "≡", you get the Boolean selection Table 10.

Table 10 All the possibilities of the RMDPL-criterion when the z-move is impossible

Dec Index	Bd23	Bd21	Bd31	Selection	Comment
0	False	False	False	$\mathbf{P}_1$	$\rho_2^2 > \rho_3^2 > \rho_1^2$
1	False	False	True	$\mathbf{P}_3$	$\rho_2^2 > \rho_1^2 > \rho_3^2$
2	False	True	False	DC $\mathbf{P}_2$ $\mathbf{P}_3$	Impossible → Don't Care a). Bd21 & Bd31=> $\mathbf{P}_2$ b). Bd23 & Bd21=> $\mathbf{P}_3$
3	False	True	True	$\mathbf{P}_3$	$\rho_1^2 > \rho_2^2 > \rho_3^2$
4	True	False	False	$\mathbf{P}_1$	$\rho_3^2 > \rho_2^2 > \rho_1^2$
5	True	False	True	DC $\mathbf{P}_1$ $\mathbf{P}_1$	Impossible → Don't Care a). Bd23 & Bd21=> $\mathbf{P}_1$ b). Bd23 & Bd21=> $\mathbf{P}_1$
6	True	True	False	$\mathbf{P}_2$	$\rho_3^2 > \rho_1^2 > \rho_2^2$
7	True	True	True	$\mathbf{P}_2$	$\rho_1^2 > \rho_3^2 > \rho_2^2$
E0	Equal	False	False	$\mathbf{P}_1$	$\rho_2^2 \geq \rho_3^2 > \rho_1^2$
E1	Equal	False	True	DC $\mathbf{P}_1$	impossible
E2	Equal	True	False	DC $\mathbf{P}_2$	impossible
E3	Equal	True	True	$\mathbf{P}_2$	$\rho_1^2 > \rho_3^2 \geq \rho_2^2$, priority $\mathbf{P}_2$
0e	False	Equal	False	DC $\mathbf{P}_2 \parallel \mathbf{P}_3$	Impossible see index 2
1e	False	Equal	True	$\mathbf{P}_3$	$\rho_2^2 \geq \rho_1^2 > \rho_3^2$
2e	True	Equal	False	$\mathbf{P}_2$	$\rho_3^2 > \rho_1^2 \geq \rho_2^2$, priority $\mathbf{P}_2$
3e	True	Equal	True	DC $\mathbf{P}_2$	impossible
0E	False	False	Equal	$\mathbf{P}_3$ $\mathbf{P}_1$	$\rho_2^2 > \rho_1^2 \equiv \rho_3^2$ by definition "≤" ≡ "<" $\Rightarrow \mathbf{P}_3$

1E	False	True	Equal	DC $\mathbf{P}_3$	impossible
2E	True	False	Equal	DC $\mathbf{P}_1$	impossible
3E	True	True	Equal	$\mathbf{P}_2$	$\rho_1^2 \geq \rho_3^2 > \rho_2^2$

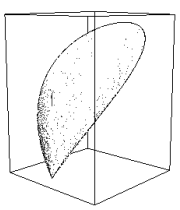
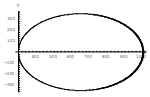
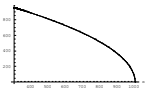
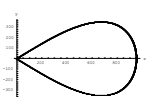
Defining all the "Equal" as "True", the table simplifies to a Karnaugh table and it means that the point $\mathbf{P}_2$ has the highest priority and that the priority of $\mathbf{P}_3$ is higher than the priority of $\mathbf{P}_1$ (normally these points have the same priority). This choice corresponds with Bd31 = Boole[$\mathbf{P}_3, \mathbf{P}_1$] ≤ 0]
A Bresenham algorithm uses only {Bd23, Bd21} and the RMDPL criterion uses {Bd23, Bd21} when Bd23 is true and {Bd21, Bd31} when Bd23 is false. Therefore, both algorithms select point $\mathbf{P}_1$ for the decimal row "5", but for the decimal row "2", the RMDPL criterion selects point $\mathbf{P}_2$ and the Bresenham algorithm selects point $\mathbf{P}_3$.
When the selection is impossible you can apply the criterion that is best adapted to the situation.
From the listofPairs (52),the listCan={ $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_1$ }, (76) –(79) and LK ≡ 0
listDV0 = { D23, D21, D31 } = { -2 LI LJ+LJ^2, LI^2-2 LI LJ, LI^2-LJ^2 }; (76)
listDVx = { d23x, d21x, d31x} = { 2 (LJ^2), -2 LI LJ, -2 (LI LJ+LJ^2) }; (77)
listDVy = { d23y, d21y, d31y} = { -2 LI LJ, 2 (LI^2), 2 (LI^2+LI LJ) }; (78)
listDVz = { d23z, d21z, d31z} = { 0, 0, 0 }. (79)
Hence the logical decision variables corresponding with point $\mathbf{P}_n$ becomes
Bd23 = Sign[D23+d23*x_n +d23*y_n];
Bd21 = Sign[D21+d21*x_n +d21*y_n];
Bd31 = Sign[D31+d31*x_n +d31*y_n];
The update is 100% incremental.
The same analysis can be done for the points { $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_6$ }, { $\mathbf{P}_3$, $\mathbf{P}_4$, $\mathbf{P}_2$ } and { $\mathbf{P}_5$, $\mathbf{P}_6$, $\mathbf{P}_4$ }.

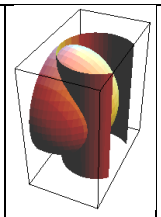
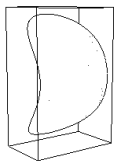
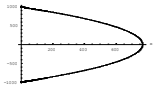
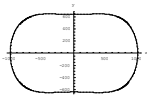
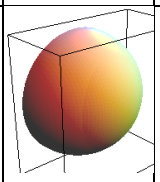
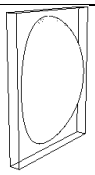
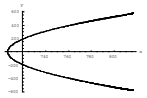
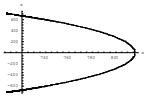
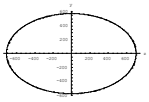
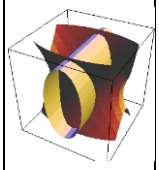
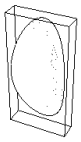
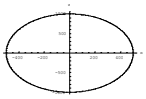
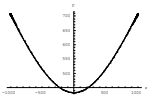
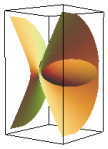
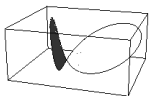
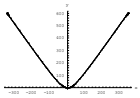
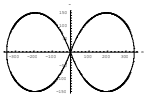
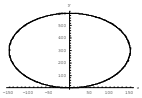
7.2 Several incremental QSICS made with the scalar decision function (47)

Table 11 Startpoints, Sign of the QSIC and initial direction vector

Example	x_s	y_s	z_s	S_L	S_x	S_y	S_z	Remarks
#4	R	0	0	-1	0	-1	+1	Singular
#8	round(R²/(2R$_1$))	round(√(R²- x_s^2))	0	-1	-1	+1	-1	zy-proj.
#10	round(R/√2)	0	round(R/√2)	-1	0	0	0	$S_{x=0}$ $S_{y=0}$ $S_{z=-0}$
#12	0	round(R/√2)	R	-1	+1	-1	-1	$S_{x=0}$ $S_{y=0}$ $S_{z=-0}$
#16	0	0	0	-1	-1	0	+1	Singular

Table 12 Examples of QSICS and its projections made by (47)

Example	Show quadrics	QSIC	xy-projection	xz-projection	zy-projection
#4: x²+y²+z²-R²=0 (x-R$_1$)²+y²-(R$_2$)²=0 R=1000 R$_1$=650 R$_2$=350					

#8: $x^2+y^2+z^2-R^2=0$ $(x-R_1)^2+y^2-(R_2)^2=0$ R=1000 R_1=650 R_2=650					
#10: $x^2+y^2+z^2-R^2=0$ $.95x^2+1.1y^2+1.05z^2-R^2=0$ R=1000					
#12: $4x^2+z^2-R^2=0$ $x^2+4y^2-z^2-R^2=0$ R=1000					
#16: $x^2+z^2-2Ry=0$ $3x^2-y^2-z^2=0$ R=100					

References

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

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

3 Huypens V., “The Sign Corrected Midpoint Decision Variable selects the candidate point with the minimum Euclidean distance to the conic”, Geometric Modeling and Imaging”, 3rd International Conference GMAI July 2008, IEEE Service Center Order Nr. P3270, ISBN 978-0-7695-3270-7, pp. 15-20, July 2008.

4 Huypens V., “Constant Speed Lines – Curves – NURBS Reference Pulse IPOs (part I] ”, https://www.researchgate.net/search

5 Huypens V., “Flowchart Berserkless Midpoint Algorithm version 4”, https://www.researchgate.net/search

6 Knuth Donald E., “The Art of Computer Programming, Bitwise Tricks & Techniques,” Volume 4, Fascicle 1, Addison-Wesley, ISBN 0-321-58050-8, 20 December 2008.

7 Chen Jihon, Kai Zhang, Yuan Zhou, Yufei Liu, Zheng Chen, Li Yin,”Exploring the Development of Research, Technology and Business of Machine Tool Domain in New-Generation Information Technology Environment Based on Machine Learning,” Sustainability 2019, 11, 3316; doi: 10.3390/su11123316.

8 Liu X-W, K. Cheng, ”Three-dimensional extension of Bresenham’s algorithm and its application in straight-line interpolation,” Proc Instn Mech Engrs, J. engineering Manufacture, 216 Part B: 45963, November 2001.

9 Cohen-Or D., Kaufman A., "3D line voxelization and connectivity control," IEEE Computer Graphics and Applications, vol. 17, no. 6, pp. 80-87, Nov.-Dec. 1997, doi: 10.1109/38.626973

10 Wang J., S. Xiao, T. Song, J. Yue, P. Bian, Yu Li, “Spatial Straight-Line Drawing Algorithm Based on Method of Discriminate Regions—A Control Algorithm of Motors”, Energies 2020, 13, 5002, doi: 10.3390/en13195002.

11 Papaioannou Spiros G., M. M. Patrikoussakis, “Curve interpolation based on the canonical arc length parametrization”, Computer-Aided Design, Vol. 43, pp. 21-30, 2011, Elsevier.

12 Kaufman A. and Shimony E., ”3D Scan-Conversion Algorithms for Voxel-Based Graphics, “Proc. 1986 Workshop on Interactive 3D Graphics, ACM Press, New York, 1986, pp 45-75

13 Bresenham J.E., “Algorithm for Computer Control of a digital plotter,” IBM Systems Journal, Vol. 4, pp. 25-30, 1965

14 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.

15 Zimenko K.V., 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.

16 Abdel-Malek K., H.J. Yeh, “Determining intersection curves between surfaces of two solids”, Computer-Aided Design, Vol. 28, No. 6/7, pp. 539-549, 1996, Elsevier Science.

17 Gonzalez-Vega L., A. Trocado, “Tools for analysing the intersection curve between two quadrics through projection and lifting”, Journal of Computational and Applied Mathematics, 11th February 2021

18 Wang W., B. Joe, T. Goldman, “Computing quadric surface intersections based on an analysis of plane cubic curves”, Graphical Models, Volume 64, Issue 6, November 2002, pages 335-367Graphics Models, 64, 2003, pp. 335-367, Example 5.1.

19 Wang W., R. Goldman, C. Tu, “Enhancing Levin’s method for computing quadric-surface intersections”, Computer Aided Geometric Design, Vol. 20, 2003, pp. 401-422, Elsevier.

20 Tu Changhe, Wenping Wang, Bernard Mourrain, Jiaye Wang, “Signature Sequence of Intersection Curve of Two Quadrics for exact Morphological Classification”, HKU C.S. Tech Report TR-2005-09, pp 1-37.

21 Tu Change, Wenping Xang, Jiaye Wang,” Classifying the nonsingular intersection curve of two quadric surfaces Theory and applications”, GMP ’02, 2002 IEEE, pg. 23-32.

22 Farouki R.T., C.A. Neff, M.A. O’Conner, ”Automatic Parsing of degenerate quadric)surface intersections”, ACM Transactions on Graphics, Vol. 8, No. 3, July 1989, Pages 174-203.

23 Bajaj C.I., C.M. Hoffmann, R.E. Lynch, J.E.H. Hopcroft, ”Tracing surface intersections”, Computer Aided Geometric Design 5, 1988, pg. 285-307, Elsevier Science Publishers B.V.

24 Hoffmann C.M., ”Algebraic Curves”, Computer Science Department, Purdue University.

25 Miller J.R. and Goldman R.N., “Combining Algebraic Rigor with geometric robustness for detection and calculation of conic sections in the intersection of two natural quadric surfaces”, ACM Symposium on Solid Modeling and Applications, Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, Austin, Texas, United States, Pages: 221 - 231, Year of Publication:

26 Dupont Laurent, Sylvain Lazard, Daniel Lazard, Sylvain Petitjean, “Near-Optimal Parameterization of the Intersection of Quadrics: Theory and Implementation”, Proc. Of SoCG (ACM Symposition on Computational Geometry), San Diego, pages 246-255, 2003.

27 Lazard Sylvain, Luis Mariano Peñaranda, Sylvain Petitjean, “Intersecting Quadrics: An Efficient and Exact Implementation”, Proc. Of the twentieth annual symposium on Computational Geometry, June 8-11, 2004, Brooklyn, New York, USA, pp 418-428

28 Dupont M Laurent, Daniel Lazard, Sylvain Lazard, Sylvain Petitjean, “Near-Optimal Parameterization of the Intersection of Quadrics: I. The Generic Algorithm”, INRIA, Rapport de recherche, Nr. 5667, September 2005. pp. 1-39.

29 Dupont L., D. Lazard, S. Lazard, S. Petitjean, “Near-Optimal Parameterization of the Intersection of Quadrics: II. A Classification of Pencils”, J. of Symbolic Computation, Vol. 43, No. 3, 2008, pp. 192-215, doi.org/10.1016/j.jsc.2007.10.012, Elsevier.

30 Van Aken J., Novak M., "Curve-Drawing Algorithms for Raster Displays," ACM Transactions on Graphics, Vol. 4, No. 2, April 1985, pp. 147-169.

31 Suh Suk-Hwan, 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.

32 Sederberg T., R. Goldman, H. Du, “Implicitizing Rational Curves by the Method of Moving Algebraic Curves”, J. Symbolic Computation, 23, pp 153-173, 1997.

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

34 Bresenham J.E. et. al., “Theoretical Foundations of Computer Graphics and CAD”, NATO ASI Series, Series F: Computer and Systems Sciences, Vol. 40, 1988, Springer-Verlag.

35 Huypens V., “Perfect 3D-line.cdf”, DOI:10.13140/RG.2.2.16382.92483, www.researchgate.net.

36 Huypens V., “Perfect 3D-curve IPO; example h1&g1.cdf”, DOI: 10.13140/RG.2.2.15800.26888, www.researchgate.net

37 Huypens V., “Bresenham’s 3D-curve IPO; example h1&g1.cdf”, DOI: 10.13140/RG.2.2.,14694.34889, www.researchgate.net

38 Omirou S. L., “Space curve interpolation for CNC machines”, Journal of Materials Processing Technology, 141, 2003, pp. 343-350.

39 Danielsson Per, “Incremental Curve Generation”, IEEE Tr. On Computers, vol. C-19, pp. 783-793, Sept. 1970.