Journal of Software for Algebra and Geometry

```
                         gap> g:= SymmetricGroup( 4 );
                         Sym( [ 1 .. 4 ] )
i5 : betti(t,Weights=>{1,0})
                         gap> tbl:= CharacterTable( g );;  HasIrr( tbl );
                         false
           0 1  2  3 4   gap> tblmod2:= CharacterTable( tbl, 2 );
o5 = total: 1 4 13 14 4  BrauerTable( Sym( [ 1 .. 4 ] ), 2 )
       0: 1 .  .  . .    gap> tblmod2 = CharacterTable( tbl, 2 );
       1: . 2  2  4 2    true
       2: . 2  5  6 .    gap> tblmod2 = BrauerTable( tbl, 2 );
       3: . .  4  . 2    true
       4: . .  .  4 .    gap> tblmod2 = BrauerTable( g );
       5: . .  2  . .    true
o5 : BettiTally          gap> tblmod2 = BrauerTable( g );
i6 : betti(t,Weights=>{0,1})
                         gap> libtbl:= CharacterTable( "M" );
           0 1  2  3 4   CharacterTable( "M" )
o6 = total: 1 4 13 14 4  gap> CharacterTableRegular( libtbl, 2 );
       0: 1 .  .  . .    BrauerTable( "M", 2 )
       1: . 2  2  4 .    gap> BrauerTable( libtbl, 2 );
       3: . 2  4  . 2    fail
       4: . .  .  4 .    gap> CharacterTable( "Symmetric", 4 );
       5: . .  2  . .    CharacterTable( "Sym(4)" )
o6 : BettiTally          gap> ComputedBrauerTables( tbl );
i7 : t1 = betti(t,Weights=>{1,1})
                         [ , BrauerTable( Sym( [ 1 .. 4 ] ), 2 ) ]
           0 1  2  3 4
o7 = total: 1 4 13 14 4
       0: 1 .  .  . .
       1: . .  .  . .
       2: . .  .  . .
       3: . 2  .  . .
       4: . .  .  . .
       5: . 2  .  . .
       6: . .  1  . .
       7: . .  8  6 .
       8: . .  4  8 4
o7 : BettiTally
i8 : peek t1

o8 = BettiTally{(0, {0, 0}, 0) => 1 }
               (1, {2, 2}, 4) => 2
               (1, {3, 3}, 6) => 2
               (2, {3, 7}, 10) => 2
               (2, {4, 4}, 8) => 1
               (2, {4, 5}, 9) => 4
               (2, {5, 4}, 9) => 4
               (2, {7, 3}, 10) => 2
               (3, {4, 7}, 11) => 4
               (3, {5, 5}, 10) => 6
               (4, {5, 7}, 12) => 2
               (4, {7, 5}, 12) => 2
```

```
ring r1 = 32003,(x,y,z),ds;
int a,b,c,t=11,5,3,0;
poly f = x^a+y^b+z^(3*c)+x^(c+2)*y^(c-1)+x^
         x^(c-2)*y^c*(y^2+t*x)^2;
option(noprot);
timer=1;
ring r2 = 32003,(x,y,z),dp;
poly f=imap(r1,f);
ideal j=jacob(f);
vdim(std(j));
==> 536
vdim(std(j+f));
==> 195
timer=0;  // reset timer
```

Algebra and Geometry

Computing symmetric determinantal representations

JUSTIN CHEN AND PAPRI DEY

Computing symmetric determinantal representations

JUSTIN CHEN AND PAPRI DEY

ABSTRACT: We introduce the `DeterminantalRepresentations` package for *Macaulay2*, which computes definite symmetric determinantal representations of real polynomials. We focus on quadrics and plane curves of low degree (i.e., cubics and quartics). Our algorithms are geared towards speed and robustness, employing linear algebra and numerical algebraic geometry, without genericity assumptions on the polynomials.

1. INTRODUCTION. The problem of representing a polynomial as the determinant of a linear matrix pencil is classical; see [Beauville 2000; Buckley and Košir 2007; Dickson 1921; Dixon 1902; Helton and Vinnikov 2007]. A polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$ of degree d (not necessarily homogeneous) is called *determinantal* if f is the determinant of a matrix with linear entries; i.e., there exist matrices $A_0, \ldots, A_n \in \mathbb{R}^{d \times d}$ such that $f(x_1, \ldots, x_n) = \det(A_0 + x_1 A_1 + \cdots + x_n A_n)$. The matrix $A_0 + x_1 A_1 + \cdots + x_n A_n$ is said to give a *determinantal representation* of f of size d. If the matrices A_i can be chosen to be all symmetric (resp., hermitian), then the determinantal representation is called *symmetric* (resp., hermitian). The determinantal representation is called *definite* if A_0 is definite, and *monic* if $A_0 = I_d$ is the identity matrix.

Computing definite symmetric (resp., hermitian) determinantal representations of a polynomial is known as the determinantal representation problem in convex algebraic geometry [Parrilo 2013]. It has generated interest in the optimization community due to its connection with the problem of determining definite linear matrix inequality (LMI) representable sets ([Helton and Vinnikov 2007; Vinnikov 2012]). The problem of characterizing the LMI-representable subsets of $\mathbb{R}^n$ (i.e., spectrahedra) can be solved by characterizing determinantal polynomials, which leads to the generalized Lax conjecture [Lewis et al. 2005].

Throughout this article we focus mainly on homogeneous polynomials, typically in three variables, corresponding to projective plane curves (though internally via dehomogenization, it suffices to compute determinantal representations for bivariate polynomials). By a celebrated theorem of Helton and Vinnikov [2007] (see also [Lewis et al. 2005]), all hyperbolic polynomials in three variables admit definite symmetric determinantal representations. When $n \geq 4$, a general homogeneous polynomial of degree d in n variables does not admit any determinantal representation of size d (except for $(n, d) = (4, 3)$). We abbreviate the terms "monic symmetric (resp., hermitian) determinantal representation" to MSDR (resp., MHDR).

By default, we assume all polynomials are defined over $\mathbb{R}$, computations are done in floating point to some fixed precision (by default 53 bits in [Macaulay2]), and all determinantal representations are definite and symmetric.

2. QUADRATIC DETERMINANTAL POLYNOMIALS.

For a quadratic polynomial $f(\boldsymbol{x}) = \boldsymbol{x}^T A \boldsymbol{x} + b^T \boldsymbol{x} + 1 \in \mathbb{R}[\boldsymbol{x}]$ in n variables, one can give a necessary and sufficient condition for an MSDR (resp., MHDR) to exist, via the associated matrix $W := A - \frac{1}{4} bb^T$; namely, one (or both) of the following conditions holds:

(1) W is negative semidefinite with rank $W \le 3$

(2) A is negative semidefinite.

One can explicitly find an MSDR (resp., MHDR) of size 2 in case (1), and one of size at most $n + 1$ in case (2); see ([Dey and Pillai 2018], Theorem 3.4) for an algorithm. This is implemented as follows:

```
Macaulay2, version 1.13

i1 : needsPackage "DeterminantalRepresentations"

i2 : R = RR[x_1..x_4]

i3 : f = -25*x_1^2 + 254*x_1*x_2 + 243*x_2^2 + 234*x_1*x_3 + 494*x_2*x_3 + 247*x_3^2
         + 198*x_1*x_4 + 378*x_2*x_4 + 378*x_3*x_4 + 143*x_4^2 + 18*x_1 + 32*x_2 + 32*x_3
         + 24*x_4 + 1

i4 : detRep f

o4 = {| 8.06514x_1+19.0421x_2+17.2128x_3+12.4043x_4+1 10.2531x_1+1.93541x_2+2.74393x_3+... |}
     | 10.2531x_1+1.93541x_2+2.74393x_3+.914643x_4   9.93486x_1+12.9579x_2+14.7872x_3+... |

i5 : clean(1e-12, f - det first oo)

o5 = 0
```

As seen above, the output of `detRep` is a list of matrices M, whose entries are linear forms, such that the input polynomial f is equal to $\det M$.

3. GENERALIZED MIXED DISCRIMINANT.

The relations between the coefficients of a determinantal polynomial and the entries of the coefficient matrices are captured by the generalized mixed discriminant:

Definition. Let $\{A^{(1)}, \dots, A^{(n)}\} = \{(a_{ij}^{(1)}), \dots, (a_{ij}^{(n)})\}$ be a set of $n \times n$ matrices of size n. The generalized mixed discriminant of a tuple of matrices

$$(\underbrace{A^{(1)}, \dots, A^{(1)}}_{k_1}, \underbrace{A^{(2)}, \dots, A^{(2)}}_{k_2}, \dots, \underbrace{A^{(n)}, \dots, A^{(n)}}_{k_n})$$

is defined as

$$\widehat{D}(\underbrace{A^{(1)}, \dots, A^{(1)}}_{k_1}, \underbrace{A^{(2)}, \dots, A^{(2)}}_{k_2}, \dots, \underbrace{A^{(n)}, \dots, A^{(n)}}_{k_n}) := \sum_{\alpha \in S[m]} \sum_{\sigma \in \widetilde{S}} \det \begin{bmatrix} a_{\alpha_1 \alpha_1}^{(\sigma(1))} & \cdots & a_{\alpha_1 \alpha_k}^{(\sigma(1))} \\ & \vdots & \\ a_{\alpha_k \alpha_1}^{(\sigma(k))} & \cdots & a_{\alpha_k \alpha_k}^{(\sigma(k))} \end{bmatrix},$$

where m is the number of distinct matrices, $S[m]$ is the set of order-preserving m-cycles in S_n (i.e., $\alpha = (\alpha_1, \dots, \alpha_m) \in S[m] \Rightarrow \alpha_1 < \alpha_2 < \cdots < \alpha_m$), and $\widetilde{S}$ is the set of all distinct permutations of

$$\{\underbrace{1, \dots, 1}_{k_1}, \dots, \underbrace{n, \dots, n}_{k_n}\}.$$

Theorem 3.1 (see [Dey 2019, Theorem 2.5]). *If $f = \det(I_d + \sum x_i A_i)$ is determinantal of degree d, then the coefficients of f are given by the generalized mixed discriminants of the matrices A_i; namely, the coefficient of $x_1^{k_1} \cdots x_n^{k_n}$ in f is equal to*

$$\widehat{D}(\underbrace{A_1, \ldots, A_1}_{k_1}, \underbrace{A_2, \ldots, A_2}_{k_2}, \ldots, \underbrace{A_n, \ldots, A_n}_{k_n}).$$

Generalized mixed discriminants can computed using `generalizedMixedDiscriminant`:

```
i6 : n = 3; S = RR[a_(1,1)..a_(n,n),b_(1,1)..b_(n,n),c_(1,1)..c_(n,n)]; R = S[x_1..x_n]
i9 : A = sub(genericMatrix(S, n, n), R)
i10 : B = sub(genericMatrix(S, b_(1,1), n, n), R)
i11 : C = sub(genericMatrix(S, c_(1,1), n, n), R)
i12 : G = generalizedMixedDiscriminant({A, B, C})
o12 = a   b   c    - a   b   c    - a   b   c    + a   b   c    - a   b   c    + ...
       3,3 2,2 1,1     3,2 2,3 1,1     2,3 3,2 1,1     2,2 3,3 1,1     3,3 2,1 1,2
i13 : P = det(id_(R^n) + x_1*A + x_2*B + x_3*C);
i14 : G == (last coefficients(P, Monomials => {x_1*x_2*x_3}))_(0,0)
o14 = true
```

4. HIGHER DEGREE DETERMINANTAL POLYNOMIALS. We now consider polynomials of degree > 2. Recall the Helton–Vinnikov theorem:

Theorem 4.1 [Helton and Vinnikov 2007]. *If $\tilde{f} \in \mathbb{R}[x_0, x_1, x_2]_d$ is hyperbolic with respect to $e = (e_0, e_1, e_2) \in \mathbb{R}^3$, then there exist real symmetric matrices $A_0, A_1, A_2 \in \mathrm{Sym}_{d \times d}(\mathbb{R})$ such that $e_0 A_0 + e_1 A_1 + e_2 A_2$ is positive-definite and*

$$\tilde{f} = \det(x_0 A_0 + x_1 A_1 + x_2 A_2).$$

Following [Plaumann et al. 2012], the polynomials satisfying Theorem 4.1 define so-called *Helton–Vinnikov plane curves*. Suppose now that $\tilde{f}$ is hyperbolic with respect to $e \in \mathbb{R}^3$. Note that if $A \in GL_3(\mathbb{R})$, then $\tilde{f} \circ A$ is hyperbolic with respect to $A^{-1}e$. Thus we may choose a general linear change of coordinates A such that $A^{-1}e = (1, 0, 0)$, and suppose that $\tilde{f}$ is hyperbolic with respect to $(1, 0, 0)$. As per Theorem 4.1, this means that A_0 is positive-definite, and furthermore we may assume $A_0 = I_d$ is the identity. Then in particular, the coefficient of x_0^d in $\tilde{f}$ is nonzero, so the homogenization of an MSDR for $\tilde{f}|_{x_0=1}$ gives an SDR for $\tilde{f}$.

Now, given a (not necessarily homogeneous) polynomial $f \in \mathbb{R}[x_1, x_2]$ of total degree d, we seek to compute $A_1, A_2 \in \mathrm{Sym}_{d \times d}(\mathbb{R})$ such that $f = \det(I_d + x_1 A_1 + x_2 A_2)$. Notice that although a nonhomogeneous polynomial of degree d may admit MSDRs of size $> d$, we only consider MSDRs which arise from dehomogenizing an SDR of $\tilde{f}$, which are of size d. It is easy to obtain the eigenvalues of the unknown matrices A_i: for instance, the eigenvalues of A_1 are the negative reciprocals of the roots of the univariate polynomial $f|_{x_2=0} = \det(I_d + x_1 A_1)$, and similarly for A_2 (note that this polynomial has nonzero roots).

Now, if the A_i are symmetric, then by the spectral theorem there are orthogonal matrices V_i such that $V_i^T A_i V_i = D_i$ is diagonal, with entries equal to the known eigenvalues of A_i, for $i = 1, 2$. Setting $V := V_2^T V_1$ (which is orthogonal), one has $f = \det(I_d + x_1 D_1 + x_2 V^T D_2 V) = \det(I_d + x_1 V D_1 V^T + x_2 D_2)$.

With this, one can obtain the diagonal entries of $V^T D_2 V$: it follows from Theorem 3.1 that these can be obtained by solving a linear system involving only D_1 (i.e., eigenvalues of A_1 — note the subscript) as well as coefficients of monomials in f which are linear in x_2. We note that the linear system giving rise to diagonal entries of $V^T D_2 V$ has unique solutions if and only if A_1 has distinct eigenvalues – if the solutions are nonunique, then (as we will see) we must choose a solution which is majorized by (the diagonal entries of) D_2. By a symmetrical argument, we may henceforth assume that diagonal entries of $V^T D_2 V$ and $V D_1 V^T$ are known.

It thus suffices to compute the symmetric matrix $V^T D_2 V$: we propose two methods to do so. The first, brute-force, method is to simply find the $\binom{d}{2}$ off-diagonal entries of $V^T D_2 V$ by solving a square polynomial system arising from Theorem 3.1, for which a numerical method for finding solutions has been implemented:

```
i15 : R = RR[x,y,z]
i16 : f = det(x*id_(R^4) + y*diagonalMatrix {4,3,2,1_R} + z*randomIntegerSymmetric(4,R))
i17 : sols = detRep f; #sols
o18 = 64
i19 : all(sols, M -> clean(1e-7, f - det M) == 0)
o19 = true
```

This method works well up to degree 4, but for higher degrees, even numerical methods take too long to finish.

An alternative, more theoretical, method for cubics is to note that since D_2 is known, finding $V^T D_2 V$ is equivalent to finding V, and in this case (of degree 3), it turns out that the Hadamard square of V can be determined essentially by linear algebra (see [Dey 2017, Theorem 2.14]).

Definition. If A, B are matrices of the same size, we denote their *Hadamard product* by $A \odot B$, i.e., $(A \odot B)_{ij} = A_{ij} B_{ij}$. We say that a square matrix A is *orthostochastic* if it is the Hadamard square of an orthogonal matrix, i.e., $A = V \odot V$ for some orthogonal matrix V. If $v, w \in \mathbb{R}^n$, we say that v is *majorized by* w if $\sum_{j=1}^n v_j = \sum_{j=1}^n w_j$ and $\sum_{j=1}^i (\tilde{v})_j \leq \sum_{j=1}^i (\widetilde{w})_j$ for all $i = 1, \ldots, n$, where $\tilde{v}, \widetilde{w}$ are the decreasing rearrangements of v, w.

To elaborate: in the cubic case, given the majorization conditions mentioned above, one can set up a zero-dimensional polynomial system to find the unknown entries of $V \odot V$, which consists of one cubic equation with all other equations linear, and only involves the (known) diagonal entries of $D_1, D_2, V^T D_2 V, V D_1 V^T$.

Thus to recover V, it suffices to determine all orthogonal matrices with Hadamard square equal to a given orthostochastic matrix. Given an $n \times n$ orthostochastic matrix A, there are 2^{n^2} possible matrices whose Hadamard square is A (not all of which will be orthogonal in general, though). Let

$G \cong (\mathbb{Z}/2\mathbb{Z})^n$ be the group of diagonal matrices with diagonal entries equal to ± 1. Then $G \times G$ acts on the set of orthogonal matrices whose Hadamard square is A, via $(g_1, g_2) \cdot O := g_1 O g_2$. The method `orthogonalFromOrthostochastic` computes all such orthogonal matrices, modulo the action of $G \times G$. We note the following:

Proposition 4.2. *For a general orthostochastic matrix A, there is exactly one $G \times G$-orbit of orthogonal matrices with Hadamard square equal to A.*

Proof. Since the action of $G \times G$ amounts to performing sign changes in each row and column, each $G \times G$ orbit contains a unique element with nonnegative entries in the first row and first column. If there were distinct $G \times G$ orbits for A, then there would be orthogonal matrices U, V with identical first column $u_{(i,1)} = v_{(i,1)} = \sqrt{a_{(i,1)}}$, but $u_j \neq \pm v_j$ for some column $j > 1$ (since $u_{(1,j)} = v_{(1,j)} = \sqrt{a_{(1,j)}} \geq 0$), which would impose a Zariski-closed condition on the entries of U (namely $u_1^T(u_j - v_j) = 0$), which does not hold for general A. $\square$

We illustrate this with some examples: note that when using floating point inputs, it may be necessary to adjust the value of the option `Tolerance` (default value 10^{-5}) in order to obtain useful results.

```
i20 : (A1, A2) = (randomIntegerSymmetric(3, R), randomIntegerSymmetric(3, R))

o20 = (| 12 3  18 |, | 18 16 9  |)
       | 3  12 14 |  | 16 4  11 |
       | 18 14 6  |  | 9  11 16 |

i21 : f = det(x*id_(R^3) + y*A1 + z*A2)

        3     2       2     3      2             2         2           2         3
o21 = x  + 30x y - 241x*y  - 3918y  + 38x z + 52x*y*z + 768y z - 34x*z  + 3282y*z  - 2278z

i22 : reps = detRep f

o22 = {| x+33.7014y+36.8578z  .607983z                3.51039z              |,
       | .607983z             x+9.08918y-6.37807z  -3.92965z                |
       | 3.51039z             -3.92965z               x-12.7906y+7.52028z   |
       -----------------------------------------------------------------------
       | x+33.7014y+36.8578z 1.11917z                3.2642z               |}
       | 1.11917z            x+9.08918y-6.37807z  4.02828z                 |
       | 3.2642z             4.02828z                x-12.7906y+7.52028z   |

i23 : all(reps, M -> clean(1e-9, f - det M) == 0)

o23 = true

i24 : g = x^3+7*x^2*y+16*x*y^2+12*y^3+3*x^2*z+22*x*y*z+32*y^2*z-45*x*z^2-65*y*z^2-175*z^3;

i25 : reps = detRep g

o25 = {| x+3y-7z 0         0       |}
       | 0        x+2y+5z 0        |
       | 0        0         x+2y+5z |

i26 : h = product gens R

o26 = x*y*z

i27 : detRep h
stdio:27:1:(3): error: Expected polynomial to be hyperbolic with respect to (1,0,0).
Try specifying a point with the option HyperbolicPt

i28 : detRep(h, HyperbolicPt => matrix{{1_RR},{1},{1}})

o28 = {| 3.63628z 0          0         |}
       | 0           .509382x 0        |
       | 0           0           .539882y |
```

As the second example shows, no smoothness assumptions are required — that is, the eigenvalues of the coefficient matrices A_1 and A_2 need not be distinct (as opposed to the treatment in, e.g., [Plaumann et al. 2012]). This in turn reveals information which is typically hidden to numerical methods (such as a `numericalIrreducibleDecomposition`), e.g., that one of the component lines of the plane curve defined by g above has multiplicity 2. The third example shows that even if the curve is in special position, one can specify a direction of hyperbolicity, and a general change of coordinates will internally be applied to compute a definite SDR (which therefore may not be monic in general).

5. ADDITIONAL METHODS. In the course of creating this package, various functions for working with matrices were needed which (to the best of our knowledge) were not available in *Macaulay2*. Thus a number of helper functions are included in this package, which may be of general interest to users beyond the scope of computing determinantal representations. These include: `hadamard` (for computing Hadamard products), `cholesky` (for computing the Cholesky decomposition of a PSD matrix), `companionMatrix` (which returns a matrix whose characteristic polynomial is any given univariate monic polynomial), `isOrthogonal`, `isDoublyStochastic` (for checking properties of a given matrix), `randomIntegerSymmetric`, `randomUnipotent`, `randomOrthogonal`, `randomPSD` (for generating various types of random matrices), and `realPartMatrix`, `roundMatrix` (for converting matrices from $\mathbb{C}$ to $\mathbb{R}$ and $\mathbb{Q}$).

SUPPLEMENT. The online supplement contains version 1.3.0 of `DeterminantalRepresentations`.

REFERENCES.

[Beauville 2000] A. Beauville, "Determinantal hypersurfaces", pp. 39–64 , 2000. MR Zbl

[Buckley and Košir 2007] A. Buckley and T. Košir, "Determinantal representations of smooth cubic surfaces", *Geom. Dedicata* **125** (2007), 115–140. MR Zbl

[Dey 2017] P. Dey, "Definite determinantal representations via orthostochastic matrices", preprint, 2017. arXiv

[Dey 2019] P. Dey, "Definite determinantal representations of multivariate polynomials", *J. of Algebra and Its Applications* (online publication July 2019).

[Dey and Pillai 2018] P. Dey and H. K. Pillai, "A complete characterization of determinantal quadratic polynomials", *Linear Algebra Appl.* **543** (2018), 106–124. MR Zbl

[Dickson 1921] L. E. Dickson, "Determination of all general homogeneous polynomials expressible as determinants with linear elements", *Trans. Amer. Math. Soc.* **22**:2 (1921), 167–179. MR Zbl

[Dixon 1902] A. C. Dixon, "Note on the reduction of a ternary quantic to a symmetrical determinant", *P. Cambridge Philos. Soc.* **11** (1902), 350–351. Zbl

[Helton and Vinnikov 2007] J. W. Helton and V. Vinnikov, "Linear matrix inequality representation of sets", *Comm. Pure Appl. Math.* **60**:5 (2007), 654–674. MR Zbl

[Lewis et al. 2005] A. S. Lewis, P. A. Parrilo, and M. V. Ramana, "The Lax conjecture is true", *Proc. Amer. Math. Soc.* **133**:9 (2005), 2495–2499. MR Zbl

[Macaulay2] D. R. Grayson and M. E. Stillman, "Macaulay2: a software system for research in algebraic geometry", available at http://www.math.uiuc.edu/Macaulay2.

[Parrilo 2013] P. A. Parrilo, "Semidefinite optimization", pp. 3–46 in *Semidefinite optimization and convex algebraic geometry*, edited by P. A. P. Grigoriy Blekherman and R. R. Thomas, MOS-SIAM Ser. Optim. **13**, SIAM, Philadelphia, PA, 2013. MR Zbl

[Plaumann et al. 2012] D. Plaumann, B. Sturmfels, and C. Vinzant, "Computing linear matrix representations of Helton–Vinnikov curves", pp. 259–277 in *Mathematical methods in systems, optimization, and control*, edited by H. Dym et al., Oper. Theory Adv. Appl. **222**, Springer, 2012. MR Zbl

[Vinnikov 2012] V. Vinnikov, "LMI representations of convex semialgebraic sets and determinantal representations of algebraic hypersurfaces: past, present, and future", pp. 325–349 in *Mathematical methods in systems, optimization, and control*, edited by H. Dym et al., Oper. Theory Adv. Appl. **222**, Springer, 2012. MR Zbl

JUSTIN CHEN:

jchen646@gatech.edu
School of Mathematics, Georgia Institute of Technology, Atlanta, GA, United States

PAPRI DEY:

pdbdn@missouri.edu
Department of Mathematics, University of Missouri, Columbia, MO, United States