

# TYPES OF SECOND-ORDER EQUATIONS

The purpose of these notes is to expand somewhat on Strauss's Ch. 1 §1.6. There is a little bit more happening in the topic than what he shows you, and the material of these notes may be helpful for people doing the first problem sheet.

I will number new formulas decimally here, so, *e.g.*, (1.01) is the first formula in §1 of these notes. A formula without a decimal point, *e.g.* (5), is a formula that occurs with the same number in Strauss's §1.6.

**1. Change-of-variables Relations.** We will not need to tamper with Strauss's notation of p. 29 ff., but let's review it briefly. Our linear second-order p. d. e. will have the form

$$\sum_{i,j=1}^n a_{ij}u_{x_i}u_{x_j} + \sum_{i=1}^n a_i u_{x_i} + a_0 u = 0 \quad (\text{where } a_{ij} = a_{ji} \text{ in the 2nd-order part}). \quad (5)$$

The old coordinates (the “ $x$ ”-es) and the new ones (the “ $\xi$ ”-s) will be related by the vector relation

$$\xi_k = \sum_{m=1}^n b_{km}x_m, \quad (6)$$

which is the product of the matrix  $B = (b_{ij})$  of coefficients and the column vector  $\mathbf{x} = (x_1, \dots, x_n)^T$ , yielding  $\xi = B\mathbf{x}$ . When Strauss says “convert to the new variables using the chain rule” he expresses the basic differentiation operators in the old coordinates in terms of those in the new system (this is the next, but unnumbered, formula on his p. 29):

$$\frac{\partial}{\partial x_i} = \sum_{k=1}^n \frac{\partial \xi_k}{\partial x_i} \frac{\partial}{\partial \xi_k} = \sum_{k=1}^n b_{ki} \frac{\partial}{\partial \xi_k} = \sum_{k=1}^n (B^T)_{ik} \frac{\partial}{\partial \xi_k} \quad (1.01)$$

so the old basic first-order partial-differentiation operators are written in terms of the new ones using the **transpose** of the matrix  $B$ . As Strauss then shows you at the bottom of p. 29, the effect on the old basic second-order partial-differentiation operators is that the matrix  $A$  of the coefficients in  $\sum_{i,j=1}^n a_{ij}u_{x_i}u_{x_j}$  is

replaced<sup>(1)</sup> by  $BAB^T$ . The fact that given any symmetric matrix  $B$  there exists an orthogonal matrix  $Q$  such that the matrix  $QBQ^T$  is diagonal is a well-known theorem of linear algebra: it is possible (and indeed necessary) to choose  $Q^T$  to be a matrix whose columns are orthonormal eigenvectors of  $B$ , whereupon  $QBQ^T$  appears as a diagonal matrix whose diagonal entries are the eigenvalues of  $B$ . In principle, therefore, we can always make a change of coordinates so that the matrix “ $A$ ” of coefficients of the second-order terms of (5) above is diagonal. It seems that then we should just be able to deal with the first-order terms without much work: for example, one of the problems on the first problem sheet has you write  $u = e^{\alpha x + \beta y}v$  and figure out how to adjust  $\alpha$  and  $\beta$  to make the first-order terms go away, following the example of Strauss's problem §1.6 #5 on p. 31.

If we just dismiss the problem by using the diagonalizability of  $A$ , however, we do not get a “computable” way to determine whether a concretely given constant-coefficient second-order linear differential operator is elliptic, parabolic, hyperbolic or what. Moreover, “or what” includes some “degenerate” equations<sup>(2)</sup> in which one is actually dealing with an equation of lower “dimension”—a smaller number of independent variables—than one may at first think. Distinguishing these from parabolic equations in concrete cases can be a little subtle, so we want to look at something that Strauss doesn't bring up.

---

<sup>(1)</sup> Note that Strauss writes the transpose operation with a superscript lower-case “ $t$ ” to the left, rather than to the right, of the name  $B$  of the matrix. I am sure this notation will have mystified some of you, because the first time I tried to read it I couldn't see what the transpose of  $A$ , which was a symmetric matrix anyhow, had to do with anything.

<sup>(2)</sup> This is an unfortunate choice of language, but we seem to be stuck with it.

It will be a help to have “nice” conversion formulas that tell us what a first-order operator written in terms of the old partial derivatives looks like in the new coordinates. The reason is that we will want to make some choices of new coordinates with a view to simplifying sets of first-order operators. The calculation is easy: if we have  $\sum_{j=1}^n c_j \frac{\partial}{\partial x_j}$  then plugging in (1.01) will give

$$\sum_{j=1}^n c_j \frac{\partial}{\partial x_j} = \sum_{j=1}^n c_j \left[ \sum_{k=1}^n (B^T)_{jk} \frac{\partial}{\partial \xi_k} \right] = \sum_{k=1}^n \left[ \sum_{j=1}^n (B^T)_{jk} c_j \right] \frac{\partial}{\partial \xi_k}. \quad (1.02)$$

If we have  $n$  first-order operators, indexed by  $1 \leq i \leq n$  so the  $i$ -th operator is  $\sum_{j=1}^n c_{ij} \frac{\partial}{\partial x_j}$ , then the same formula will read

$$\sum_{j=1}^n c_{ij} \frac{\partial}{\partial x_j} = \sum_{k=1}^n \left[ \sum_{j=1}^n c_{ij} (B^T)_{jk} \right] \frac{\partial}{\partial \xi_k} \quad (1.03)$$

where we recognize the coefficients of the  $\frac{\partial}{\partial \xi_k}$ 's as the matrix  $CB^T$ , where  $C$  is the matrix whose  $i$ -th row is the row of coefficients in the  $i$ -th differential operator. As we shall see, picking  $B$  in such a way as to make this matrix diagonal<sup>(3)</sup> is a way to determine the type of an equation also, and there is a reasonably “algorithmic” solution to this problem that does not require the finding of eigenvalues, at least in the most important cases.

**2. Completing the Square, AKA the  $LDL^T$  Factorization: Some Low-Dimensional Examples.** As an example, let us consider the homogeneous second-order linear partial differential operator

$$u \mapsto u_{xx} + 2u_{xy} + 5u_{yy}. \quad (2.01)$$

The matrix of coefficients of its second-order part (which is the only part that it has) is  $\begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}$ , the eigenvalues of which are easily seen to be irrational numbers, so the eigenvectors will also have to have some irrational entries, and while these only involve square roots (“quadratic surds”) in the 2-dimensional case, worse things could be expected in higher dimensions. On the other hand, we can “complete the square” in the operator itself quite easily:

$$\partial_x \partial_x + 2\partial_x \partial_y + 5\partial_y \partial_y = \partial_x \partial_x + 2\partial_x \partial_y + \partial_y \partial_y + 4\partial_y \partial_y = (\partial_x + \partial_y)^2 + (2\partial_y)^2. \quad (2.02)$$

If we regard the two operators  $\partial_x + \partial_y$  and  $2\partial_y$  as the first and second operators of (1.03) above, then the matrix “ $C$ ” is given by  $C = \begin{bmatrix} 1 & 1 \\ 0 & 2 \end{bmatrix}$ . The easiest (though not the only) choice of  $B$  such that  $CB^T$  is diagonal would be  $B^T = C^{-1}$ , or  $B = \begin{bmatrix} 1 & 0 \\ -1/2 & 1/2 \end{bmatrix}$ . Indeed, since that choice makes  $CB^T = I$ , tracing through the computations above shows that the operator we started with has become  $\partial_{\xi_1}^2 + \partial_{\xi_2}^2$ , the Laplace operator, in the new coordinates. {Doubters may choose to plug the equations  $\partial_x = \partial_{\xi_1} - \frac{1}{2} \partial_{\xi_2}$  and  $\partial_y = \frac{1}{2} \partial_{\xi_2}$  into the original operator; they will get the same result after somewhat laborious computation.}

---

<sup>(3)</sup> The easiest way to do this will turn out to be to make it the identity matrix.

Suppose that instead of this homogeneous second-order operator we had had an operator with the same second-order part, say

$$u \mapsto u_{xx} + 2u_{xy} + 5u_{yy} + u_x - 2u_y + 3u. \quad (2.03)$$

Then the same change of coördinates would have made it into

$$u_{\xi_1\xi_1} + u_{\xi_2\xi_2} + c_1u_{\xi_1} + c_2u_{\xi_2} + c_0u \quad (2.04)$$

where the coefficients that we indicated as  $c_1$ ,  $c_2$  and  $c_0$  are not important in the present discussion. However, much different situations obtain in the cases represented by the operators

$$\partial_x^2 - 4\partial_x\partial_y + 4\partial_y^2 - \partial_x - 2\partial_y \quad \text{and} \quad (2.05p)$$

$$\partial_x^2 - 4\partial_x\partial_y + 4\partial_y^2 - \partial_x + 2\partial_y, \quad (2.05d)$$

and those situations differ greatly from each other. Completing the square leads in the respective cases to

$$(\partial_x - 2\partial_y)^2 - \partial_x - 2\partial_y \quad \text{and} \quad (2.06p)$$

$$(\partial_x - 2\partial_y)^2 - \partial_x + 2\partial_y. \quad (2.06d)$$

The highest-order part of the operator is the square of a single first-order operator rather than being the product of two different first-order operators or the sum or difference of the squares of two different such operators. This part of the operator gives us only half the information we need to make a useful change of coördinates. If we rewrite (2.06p) in the form  $(\partial_x - 2\partial_y)^2 - (\partial_x + 2\partial_y)$ , however,<sup>(4)</sup> we see that we need a first-order operator different from the one involved in the second-order part in order to be able to express the first-order part. The operator is therefore “essentially two-dimensional.” Using the same notation as before,

we see that if we use the coefficients of these two first-order operators to form  $C = \begin{bmatrix} 1 & -2 \\ 1 & 2 \end{bmatrix}$  and change

coördinates in such a way that  $B^T = C^{-1}$ , *i.e.*,  $B = \begin{bmatrix} 1/2 & -1/4 \\ 1/2 & 1/4 \end{bmatrix}$ , then this operator will transform into

$$u \mapsto \partial_{\xi_1}^2 u - \partial_{\xi_2} u, \quad (2.07p)$$

which—except for the eccentric choice of the names of the coördinates—is the heat or diffusion operator  $\partial_x^2 - \partial_t$ , the study of whose solutions will occupy much of our time this semester. In the case of the operator (2.05d) (equivalently, (2.06d)), however, the equation contains no other first-order operator for us to work with. If we are to find a change-of-coördinates matrix  $B$ , then we shall have to come up with another (linearly independent) first-order operator to give us the matrix “ $C$ .” We can try using our previous choice (because in fact the choice of the second operator will be irrelevant): the operator (2.06d) will now transform into  $u \mapsto \partial_{\xi_1}^2 u - \partial_{\xi_1} u$ . This operator is very different from the heat/diffusion operator: it involves only one independent variable  $\xi_1$ , with  $\xi_2$  entering only as a “parameter,” *i.e.*, the operator involves no differentiations with respect to  $\xi_2$ , and we are dealing only with an ordinary, or one-dimensional, d. e. Indeed, we could proceed to solve  $\partial_{\xi_1}^2 - \partial_{\xi_1} = 0$  without any trouble: the solutions are of the form  $u(\xi_1, \xi_2) = C_0(\xi_2) + C_2(\xi_2)e^{\xi_1}$  because we know how to solve the o. d. e.  $f'' - f' = 0$ . Transformed back to the original  $(x, y)$ -coördinates, this solution has the form

$$u(x, y) = C_1 \left( \frac{x}{2} + \frac{y}{4} \right) + C_2 \left( \frac{x}{2} + \frac{y}{4} \right) \exp \left( \frac{x}{2} - \frac{y}{4} \right) = f(2x + y) + g(2x + y) \cdot \exp \left( \frac{x}{2} - \frac{y}{4} \right) \quad (2.07d)$$

and it is easy to check that functions of this form satisfy the original equation. The heat/diffusion equation (2.06), of course, admits no such facile “general solution,” as will become clear as the course progresses.

---

<sup>(4)</sup> The change of sign wasn't really essential in this computation: we just made it so that the formulas would look more familiar.

For second-order constant-coefficient linear partial differential operators in two variables, then, the problem of determining types is not very complicated. Start with a general operator

$$u \mapsto a_{11}\partial_x^2 u + 2a_{12}\partial_x\partial_y u + a_{22}\partial_y^2 u + a_1\partial_x u + a_2\partial_y u + a_0 u . \quad (2.08)$$

If  $a_{12} = 0$  (*i.e.*, there is no  $\partial_x\partial_y$  term) then, up to a change of scale, the second-order part looks like the Laplace operator, the (1-dimensional) wave operator, or the second-partial-derivative operator in only one of the variables: in the first case the operator is **elliptic**, in the second **hyperbolic**, while the third possibility requires further analysis. If  $a_{11} = 0 = a_{22}$  (*i.e.*, the  $\partial_x\partial_y$  term is the only one present) then with a suitable change of coordinates the operator again looks like the 1-dimensional wave operator and is consequently **hyperbolic**: one can see this from the identity

$$\partial_x\partial_y = \frac{1}{4}(\partial_x + \partial_y)^2 - \frac{1}{4}(\partial_x - \partial_y)^2 . \quad (2.09)$$

In the cases that remain, one can complete the square as we did in the examples worked out above. If two linearly independent first-order operators are involved after completing the square in (2.08), then a suitable change of coordinates (which can be determined effectively if the coefficients are known constants) will bring the second-order part into the form of either the Laplace operator (sum of squares, **elliptic** operator) or the wave operator (difference of squares, **hyperbolic** operator). If the second-order part of (2.08) is the square of a first-order operator (as would also be the case if  $a_{12} = 0$  and one or the other of  $a_{11}$  or  $a_{22}$  is 0) then one considers the first-order part of the operator (2.08). If that is linearly independent of the operator in the second-order part, then a suitable change of coordinates will bring the first- and second-order parts of the operator (2.08) into the form  $A\partial_{\xi_1}^2 + B\partial_{\xi_2}$ , and one will be dealing with an operator that is essentially the heat/diffusion operator (perhaps plus a 0-th-order term), *i.e.*, one has a **parabolic** operator. In the only remaining case the operator can be brought into the form  $u \mapsto A\partial_{\xi_1}^2 u + B\partial_{\xi_1} u + Cu$ , and one will be dealing with an operator that is essentially one-dimensional—one of those **degenerate** operators.

**3. The General, Higher-Dimensional Case.** We have not really practiced truth in labeling here: because of complications like the “ultrahyperbolic” cases, we shall pretty much restrict our attention to elliptic and parabolic cases. The first thing one needs is the higher-dimensional, or general, formulation of the process of completing the square.<sup>(5)</sup> This is very similar to Gaussian elimination, at least in the cases we shall consider. Consider the (symmetric) matrix  $A$  of coefficients of the second-order part of an operator like Strauss’s (5):

$$u \mapsto \sum_{i,j=1}^n a_{ij}u_{x_i}u_{x_j} \quad (\text{where } a_{ij} = a_{ji}). \quad (3.01)$$

The Gaussian-elimination algorithm (without pivoting) applied to a matrix  $A$  is equivalent to left-multiplication of  $A$  by successive elementary matrices that perform row reduction on  $A$ . In the present context (with symmetric  $A$ ), consider what happens if, whenever you perform a row operation on  $A$  by the elementary matrix  $E$ , you also perform the corresponding column operation on  $A$  by right-multiplication by  $E^T$ . Then you replace  $A$  by  $EAE^T$ , which is still symmetric (it equals its transpose). Successively performing these operations will (unless a zero occurs on the diagonal) eventually result in a diagonal matrix (not an upper-triangular one, because one has been removing the super-diagonal elements of  $A$  also). Here’s a simple case,

in which  $A = \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix}$  is exactly the matrix of coefficients in (2.01) above:

$$\begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} . \quad (3.02)$$

---

<sup>(5)</sup> A process very much like this one is called *Cholesky factorization* of  $A$ , and readers who are EE’s may have seen that process in a different context. The matrix “ $B$ ” formed by the process we describe here is the inverse of the matrix “ $L$ ” that Cholesky factorization produces in writing  $A=LL^T$ . An important difference, however, is that the matrix  $A$  does not have to be invertible in the algorithm presented here.

Rewritten as  $A = E^{-1}D(E^T)^{-1}$  where  $D$  is the diagonal matrix on the r. h. side, (3.02) says from the standpoint of this differential operator that

$$\begin{aligned} \partial_x \partial_x + 2\partial_x \partial_y + 5\partial_y \partial_y &= [\partial_x \ \partial_y] A \begin{bmatrix} \partial_x \\ \partial_y \end{bmatrix} = [\partial_x \ \partial_y] \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \partial_x \\ \partial_y \end{bmatrix} \\ &= [\partial_x + \partial_y \ \partial_y] \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} \partial_x + \partial_y \\ \partial_y \end{bmatrix} = (\partial_x + \partial_y)^2 + 4\partial_y^2. \end{aligned} \quad (3.03)$$

What we're seeing is that this "two-sided Gaussian elimination" does the same algebra on the matrix that completing-the-square does on the second-order partial differential operator. Indeed, one could even get rid of the "4" in the (2,2) position in the matrix: by dividing the second row (and then column) by  $\sqrt{4} = 2$ , one could further modify (3.02) to get

$$\begin{bmatrix} 1 & 0 \\ -1/2 & 1/2 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & -1/2 \\ 0 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (3.04)$$

although it is not necessary to do that to see that the operator is an elliptic operator. The fact that the matrix  $\begin{bmatrix} 1 & 0 \\ -1/2 & 1/2 \end{bmatrix}$  is the same as the "B" that gave us the change of coordinates for this operator back on p. 2 above is no accident, and we shall examine the reason for this in general, once we have the general "completing-the-square" algorithm set up.

An adequate "hand-operated" algorithm in the  $n$ -dimensional setting for keeping track of the result of successive row operations like the one we performed in (3.02) is the following: Produce a partitioned matrix  $[A|I_n]$ , the same one you produce in the familiar<sup>(6)</sup> algorithm for hand computation of  $A^{-1}$ . Do row reduction (**without pivoting**—but see the exception below) on this partitioned matrix as if you were computing  $A^{-1}$ , **but** every time you perform a row operation (whether it is that of adding/subtracting one row to/from another or multiplying/dividing a row by a nonzero constant), **perform the same column operation on the corresponding column of  $A$ , leaving the right-hand half of the partitioned matrix alone.** Thus every time you multiply  $A$  on the left by an elementary matrix  $E$  (and leave a record of the operation in the right-hand half of the partitioned matrix) you will also multiply it on the right by  $E^T$ , just as we did in the  $2 \times 2$  example above. Note that any diagonal element that is multiplied by, say  $\alpha$  in the  $k$ -th row must also be multiplied by  $\alpha$  in the  $k$ -th column. Thus you will be able to reduce a positive diagonal element to 1, or a negative diagonal element to  $-1$ —*e.g.*, if  $-3$  occurs at the  $k$ -th position on the diagonal you will be able to divide the  $k$ -th row and the  $k$ -th column by  $\sqrt{3}$ —but you will not be able to get rid of the minus sign (we are tacitly assuming that all the arithmetic is being done in  $\mathbb{R}$ ). The reader might try, for example,

applying the algorithm to  $A = \begin{bmatrix} 44/9 & -22/9 & -26/9 \\ -22/9 & 29/9 & 4/9 \\ -26/9 & 4/9 & 53/9 \end{bmatrix}$ ; if the diagonal elements are not normalized to

equal 1, then the final form of the partitioned matrix will be  $\begin{bmatrix} 44/9 & 0 & 0 & | & 1 & 0 & 0 \\ 0 & 2 & 0 & | & 1/2 & 1 & 0 \\ 0 & 0 & 81/22 & | & 37/44 & 1/2 & 1 \end{bmatrix}$ . (This

particular matrix  $A$  was chosen because it has integer eigenvalues (1, 4, 9) and rational eigenvectors—so it is very special indeed. The fractions may be ugly, but they were produced without having to solve a cubic equation to get the eigenvalues.) Of course, this is the kind of computation that any sensible person does in **Maple** or **MATLAB**. A really non-sensible person might want to check that this process has in fact completed the square in the operator

$$\frac{1}{9} [44\partial_x^2 - 44\partial_x \partial_y - 52\partial_x \partial_z + 29\partial_y^2 + 8\partial_y \partial_z + 53\partial_z^2]. \quad (3.05)$$

<sup>(6)</sup> Everybody learns how to do this in their first course in linear algebra, either from the textbook or from the folklore.

To do this, one needs to remember that the matrix “ $C$ ” of p. 2 above was  $(B^T)^{-1}$ . In this setting, the “ $B$ ” for which  $BAB^T$  is diagonal is the r. h. side of the partitioned matrix: the inverse of its transpose is  $\begin{bmatrix} 1 & -1/2 & -13/22 \\ 0 & 1 & -1/2 \\ 0 & 0 & 1 \end{bmatrix}$ . Such a person would therefore want to check that  $\frac{44}{9} \left( \partial_x - \frac{1}{2} \partial_y - \frac{13}{22} \partial_z \right)^2 + 2 \left( \partial_y - \frac{1}{2} \partial_z \right)^2 + \frac{81}{22} \partial_z^2 = \frac{1}{9} [44\partial_x^2 - 44\partial_x\partial_y - 52\partial_x\partial_z + 29\partial_y^2 + 8\partial_y\partial_z + 53\partial_z^2]$ , and one can only hope that s/he would use Maple or MATLAB.

This algorithm should be regarded as pedagogical rather than practical, even though it has the nice property of delivering the change-of-coordinates matrix directly. For one thing, it has the flaw that it can't even get started on a matrix like  $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ : if one admits the operation of interchanging two rows then one must also interchange the corresponding columns, but then this matrix doesn't change. This matrix has the eigenvalues 1 and  $-1$  and thus diagonalizes into  $\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ , so it belongs to a hyperbolic operator (the wave operator with  $c = 1$ , of course). If one is given an operator that one can reasonably assume is not hyperbolic (*e.g.*, because of a physical interpretation), however, things are a little better. If we look at the classification of the operators, we see that the elliptic and parabolic cases, as well as some degenerate ones, share the property that the matrix  $A$  of coefficients in the second-order part is (symmetric and) **nonnegative semi-definite**, that is: for any vector  $\mathbf{x} \in \mathbb{R}^n$  the inner product  $\mathbf{x}^T A \mathbf{x}$  (a  $1 \times 1$  matrix that any sensible person treats as a real number) is always nonnegative. Equivalently, all the values of the polynomial  $\sum_{i,j=1}^n a_{ij} x^i x^j$  are nonnegative. Evidently any polynomial that *can* be written as a sum of squares of first-

degree polynomials has this property. Now it is an easy fact of linear algebra that the diagonal entries in such a matrix must be nonnegative, and it is an elementary but rather subtle theorem of linear algebra that the entries of a (symmetric) nonnegative semi-definite matrix  $A = [a_{ij}]$  must satisfy the inequality  $a_{ij}^2 \leq a_{ii}a_{jj}$  for any indices  $1 \leq i, j \leq n$ . Running the algorithm described above, we see that each time a matrix  $A$  is replaced by a matrix  $EAE^T$  its nonnegative-semi-definiteness is preserved. So at the  $k$ -th stage of the algorithm, one of three things happens. If the  $(k, k)$ -entry in the current working matrix is positive, one can run another step of the algorithm. If it is negative, we are not looking at a nonnegative-semi-definite matrix! If it is zero and the original matrix  $A$  was nonnegative-semi-definite, then all the entries in the  $k$ -th row and  $k$ -th column are zero: the working matrix has partitioned itself with “no action necessary” on the  $k$ -th row and column (but the choice of the matrix “ $C$ ” of the example may be harder than we think). If it is zero but some entry in the  $k$ -th row or column is *not* zero, then the original matrix was not nonnegative semi-definite and one should reassess the situation. However, it is not easy to predict the outcome of the

algorithm: for example, if the original matrix is  $A = \begin{bmatrix} 2 & 8/7 & 10/7 \\ 8/7 & 32/49 & 40/49 \\ 10/7 & 40/49 & 197/49 \end{bmatrix}$ , then after the first step of

the algorithm the partitioned matrix will be  $\left[ \begin{array}{ccc|ccc} 2 & 0 & 10/7 & 1 & 0 & 0 \\ 0 & 0 & 0 & -4/7 & 1 & 0 \\ 10/7 & 0 & 197/49 & 0 & 0 & 1 \end{array} \right]$ ; the zero at the (2,2)

position has partitioned the l. h.  $3 \times 3$  matrix, as predicted if  $A$  was nonnegative, and we can use the pivot at

(1,1) to zero the entry at (1,3), obtaining  $\left[ \begin{array}{ccc|ccc} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -4/7 & 1 & 0 \\ 0 & 0 & 3 & -5/7 & 0 & 1 \end{array} \right]$  as the final form of the partitioned

matrix. The l. h. side of the matrix  $[A|I_3]$  is partitioned by rows and columns of zero, as we predicted for a nonnegative-semi-definite  $A$ . If we had been dealing with a second-order partial differential operator that

had come to us in the form

$$2\partial_x^2 + \frac{16}{7}\partial_x\partial_y + \frac{20}{7}\partial_x\partial_z + \frac{32}{49}\partial_y^2 + \frac{80}{49}\partial_y\partial_z + \frac{197}{49}\partial_z^2, \quad (3.06)$$

then the matrix  $B_0 = \begin{bmatrix} 1 & 0 & 0 \\ -4/7 & 1 & 0 \\ -5/7 & 0 & 1 \end{bmatrix}$  would furnish a change of coördinates for which the sum-of-squares representation of (3.06) would look like

$$2\left(\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z\right)^2 + 3\partial_z^2; \quad (3.07)$$

remember that the rows of the matrix  $(B_0^T)^{-1} = \begin{bmatrix} 1 & 4/7 & 5/7 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$  give the coefficients of the first-order operators whose squares *may* appear, so it is a good “first guess” for the matrix  $C$  that will give us all the first-order operators we need, as in the example of §1 above. But here we have only two first-order operators, namely  $\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z$  and  $\partial_z$ , occurring in the second-order operator when we have written it as a sum of squares. In a concrete situation, therefore, we would have to distinguish between the types of the operators of the form given in Strauss’s formula (5) that had this  $A$  as the matrix of their second-order part, based on whether their first-order part was (1) **linearly independent** of the operators  $\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z$  and  $\partial_z$ , which we would expect to be the **parabolic** case, or (2) **linearly dependent** on those two operators, in which case we would think (or hope) that a suitable choice of coördinates would rewrite the operator with at least one “missing coördinate” and show that it was a **degenerate** operator.

Changing coördinates with the matrix  $B_0$  has given us the “basis of first-order linear operators” whose coefficients are the rows of  $(B_0^T)^{-1}$ , namely  $\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z$ ,  $\partial_y$ , and  $\partial_z$ . If our original operator had first-order part  $a_1\partial_x + a_2\partial_y + a_3\partial_z$ , then in the coördinates given by  $[\xi_1 \xi_2 \xi_3]^T = B_0[x y z]^T$  it will be the linear combination of  $\partial_{\xi_1}$ ,  $\partial_{\xi_2}$  and  $\partial_{\xi_3}$  whose coefficients are given by the  $(1 \times 3)$  row matrix  $[a_1 \ a_2 \ a_3]B_0^T$ , whose components we could compute but which we shall denote by  $[\alpha \ \beta \ \gamma]$  for the sake of generality. It will be obvious on a moment’s reflection that  $a_1\partial_x + a_2\partial_y + a_3\partial_z$  was linearly independent of  $\left\{\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z, \partial_z\right\}$

if and only if  $\beta \neq 0$ . Now because  $\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z$  transformed to  $\partial_{\xi_1}$  and  $\partial_z$  transformed to  $\partial_{\xi_3}$  in the new coördinates, the original operator will be parabolic or degenerate respectively if and only if the rows of the

matrix  $C_0 = \begin{bmatrix} 1 & 0 & 0 \\ \alpha & \beta & \gamma \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4/7 & 5/7 \\ a_1 & a_2 & a_3 \\ 0 & 0 & 1 \end{bmatrix} B_0^T$  are linearly independent, *i.e.*, if and only if  $C_0^{-1}$  exists.

$C_0^{-1}$  will exist if and only if  $\beta \neq 0$ , and it is easy to see that this condition—which in this example would be equivalent to requiring that  $a_2 \neq \frac{4}{7}a_1$ —is necessary and sufficient for one’s being able to write  $\partial_y$  in terms of  $\left\{\partial_x + \frac{4}{7}\partial_y + \frac{5}{7}\partial_z, a_1\partial_x + a_2\partial_y + a_3\partial_z, \partial_z\right\}$ . So  $C_0$  is not invertible if and only if  $[\alpha \ \beta \ \gamma] = [\alpha \ 0 \ \gamma]$ , and we can see already that in the new coördinate system furnished by  $B_0$  the operation  $\partial_{\xi_2}$  of differentiation w. r. t. the coördinate  $\xi_2$  is nowhere involved, and the operator is degenerate.

We still have to finish the case in which  $\beta \neq 0$ . We can look at this case in the following way: the change of coördinates  $B_0$  has moved the second-order part of the operator to where we want it, and since  $C_0^{-1}$  would finish the job by moving the operator  $a_1\partial_x + a_2\partial_y + a_3\partial_z$  to the operator  $\partial_{\xi_2}$  and the matrix

that rewrites the operators is the transpose of the change-of-coördinates matrix, the “corrected” change of coördinates should be given by  $B = (C_0^{-1})^T B_0$ . Note that because this matrix is actually given by

$$\begin{aligned}
 B &= (C_0^{-1})^T B_0 = \left\{ \left[ \begin{array}{ccc|c} 1 & 4/7 & 5/7 & \\ a_1 & a_2 & a_3 & B_0^T \\ 0 & 0 & 1 & \end{array} \right]^{-1} \right\}^T B_0 \\
 &= \begin{bmatrix} 1 & a_1 & 0 \\ 4/7 & a_2 & 0 \\ 5/7 & a_3 & 1 \end{bmatrix}^{-1} B_0^{-1} B_0 = \begin{bmatrix} 1 & a_1 & 0 \\ 4/7 & a_2 & 0 \\ 5/7 & a_3 & 1 \end{bmatrix}^{-1}
 \end{aligned} \tag{3.08}$$

less matrix calculation is needed here than seems to be: one takes  $B_0^{-1}$ , replaces its second column (corresponding to the fact that  $B_0 A B_0^T$  has its second row and column composed only of zeros) by  $[a_1 \ a_2 \ a_3]^T$ , and inverts the result.<sup>(7)</sup> In continuing the exposition of this example, however, we shall use the form  $B = (C_0^{-1})^T B_0$  because the effect of the computations will be clearer.

Because it is such a sparse matrix, it is easy to compute  $C_0^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\alpha/\beta & 1/\beta & -\gamma/\beta \\ 0 & 0 & 1 \end{bmatrix}$  explicitly. We

can now see the effect of replacing  $B_0$  with  $B$ :

$$\begin{aligned}
 BAB^T &= [C_0^{-1}]^T B_0 A B_0^T C_0^{-1} \\
 &= [C_0^{-1}]^T \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} C_0^{-1} \quad \text{since } B_0 A B_0^T \text{ is already known} \\
 &= \begin{bmatrix} 1 & -\alpha/\beta & 0 \\ 0 & 1/\beta & 0 \\ 0 & -\gamma/\beta & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -\alpha/\beta & 1/\beta & -\gamma/\beta \\ 0 & 0 & 1 \end{bmatrix} \quad \text{from our computation of } C_0^{-1} \\
 &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad \text{because the second row and column of } B_0 A B_0^T \text{ are zero vectors.}
 \end{aligned} \tag{3.09}$$

So in this example—in which what happens is really perfectly general—we see that the change of coördinate matrix  $B$  both makes the second-order part of the p. d. operator a sum of squares *and* turns the first-order part of the p. d. operator into a coördinate differentiation in the direction represented by the zero row and column of  $BAB^T$ .

**4. A Fairly General Procedure.** To determine the type of a second-order constant-coefficient linear p. d. operator on  $\mathbb{R}^n$ , then, one can proceed as follows. One can begin by running the algorithm described on p. 5 above. If the matrix  $A = [a_{ij}]$  of coefficients of the second partial derivatives is positive semi-definite, then the algorithm will run, possibly leaving one or more zero rows and corresponding columns in the diagonalized matrix  $B_0 A B_0^T$  but with the matrix  $B_0$  it produces invertible. While there are various tests for positive semi-definiteness of  $A$ , the easiest test is simply to run the algorithm of p. 5: if it halts because there is a zero at the  $(k, k)$ -location on the diagonal but there is some nonzero element in the  $k$ -th

---

<sup>(7)</sup> Not even the inversion is needed to get the “basic first-order operators,” of course.

row or column, then the matrix was not nonnegative semi-definite. Otherwise a diagonal matrix  $B_0AB_0^T$  results, possibly with the  $k$ -th row and column zero for one or more values of  $1 \leq k \leq n$ . It is possible for negative entries to be present on the diagonal, but their presence shows that the matrix was not negative semi-definite: the algorithm will give a diagonal matrix with all diagonal entries nonnegative *if and only if* the original matrix was nonnegative semi-definite. If all the diagonal entries of  $B_0AB_0^T$  are strictly positive, then the coordinate transformation  $\xi = B_0\mathbf{x}$  will transform the second-order part of the given operator into an operator of the form  $\sum_{j=1}^n d_j \partial_{\xi_j}^2$  with all the  $d_j > 0$  and the operator is elliptic, the lower-order parts of the operator being irrelevant in this setting. If there is exactly one index  $k$  corresponding to a zero row and zero column, then it may be possible to modify the coordinate change matrix  $B_0$  in such a way that  $B_0AB_0^T$  is unaltered but the first-order part  $\sum_{j=1}^n a_j \partial_{x_j}$  of the operator is transformed into  $\partial_{\xi_k}$ : this will be possible if and only if the matrix produced by replacing the  $k$ -th column of  $B_0^{-1}$  by  $[a_1 \cdots a_n]^T$  is invertible, and then the inverse  $B$  of that matrix effects the desired transformation of the given operator into one of the form  $\sum_{j=1}^n d_j \partial_{\xi_j}^2 + \partial_{\xi_k}$ , where all the  $d_j > 0$  except that  $d_k = 0$ ; this operator is parabolic and therefore so is the original operator. If the matrix just described is not invertible, then the operator is degenerate. That will also be the case if two or more diagonal entries of  $B_0AB_0^T$  are zero, since even if the first-order part of the given operator can be transformed into differentiation in one of the corresponding coordinate directions, the transformed operator will still contain no operation of differentiation in the remaining coordinate directions corresponding to the coordinate transformation  $\xi = B\mathbf{x}$ .

**5. Using Existing Software to Type P. D. E.'s.** It is quite easy to run the algorithm described on p. 5 ff. above in **Maple** or in the symbolic-manipulation package of **MATLAB**. Both these packages also contain a Cholesky-factorization routine, *i.e.*, a built-in command that factors a given positive-**definite** symmetric matrix  $A$  into the form  $A = LL^T$ , where  $L$  is a lower-triangular matrix with positive entries on its diagonal. It is obvious that any such  $LL^T$  must be invertible, and so both the **Maple** and **MATLAB** routines make error returns (with less-than-helpful messages) when  $A$  is only nonnegative semi-definite. However, one might as well begin by running the Cholesky routine on the matrix of coefficients of the second-order part of the operator: if no errors result, then comparing  $BAB^T = \text{diag}$  with  $A = LL^T$ , one sees that  $B = L^{-1}$  will be a change-of-coordinates matrix for which  $BAB^T = I_n$ , the identity matrix, which is about as diagonal as you can get: the p. d. operator is elliptic.

If the Cholesky routine fails, then there are still three possibilities open. Using built-in row- and column-operation routines, it is not difficult to run the algorithm of p. 5 ff. on  $A$ . It does not depend on invertibility of  $A$ , so it fails only if  $A$  is not nonnegative semi-definite, and we are excluding that case from consideration. The rank of  $B_0AB_0^T$  is the same as that of  $A$ , so if  $A$  is of rank  $n - 1$  (with only one row/column of zeros) one needs to treat the first-order part of the given operator as we did in the example worked through on pp. 7–8 above; built-in routines in **Maple** and **MATLAB** will of course do the invertibility test and compute the inverse of the relevant matrix. In **Maple**, one could begin by applying the built-in function **rank** to  $A$ ; if **rank(A) = n** then—provided that  $A$  is nonnegative definite—executing the command **L := chollesky(A)** will return the inverse of the matrix we have been calling  $B$ . If **rank(A) = n - 1** there is some hope that the operator is parabolic; of course one will have to go through the algorithm of p. 5 ff. above to determine whether the first-order part will save the day. If **rank(A) ≤ n - 2** the operator must be degenerate, and there is really no point in proceeding farther.

One must emphasize that all the material in these notes is intended pedagogically and not as “real-life” calculation. While the property of being a (symmetric) positive definite matrix is numerically stable, having rank  $< n$  is not: there are matrices of rank  $n$  arbitrarily close to any  $n \times n$  matrix. Therefore a very small perturbation of *all the coefficients* of a parabolic operator can easily make it become either elliptic or hyperbolic. Of course an equation of the form  $u_t = \sum_{i,j=1}^n a_{ij} u_{x_i} u_{x_j}$  with positive definite  $A = [a_{ij}]$ , like the heat/diffusion operator, will remain parabolic under fairly heavy perturbation in the space variables only. So the algorithmic material of these notes is presented only to help people do textbook problems.